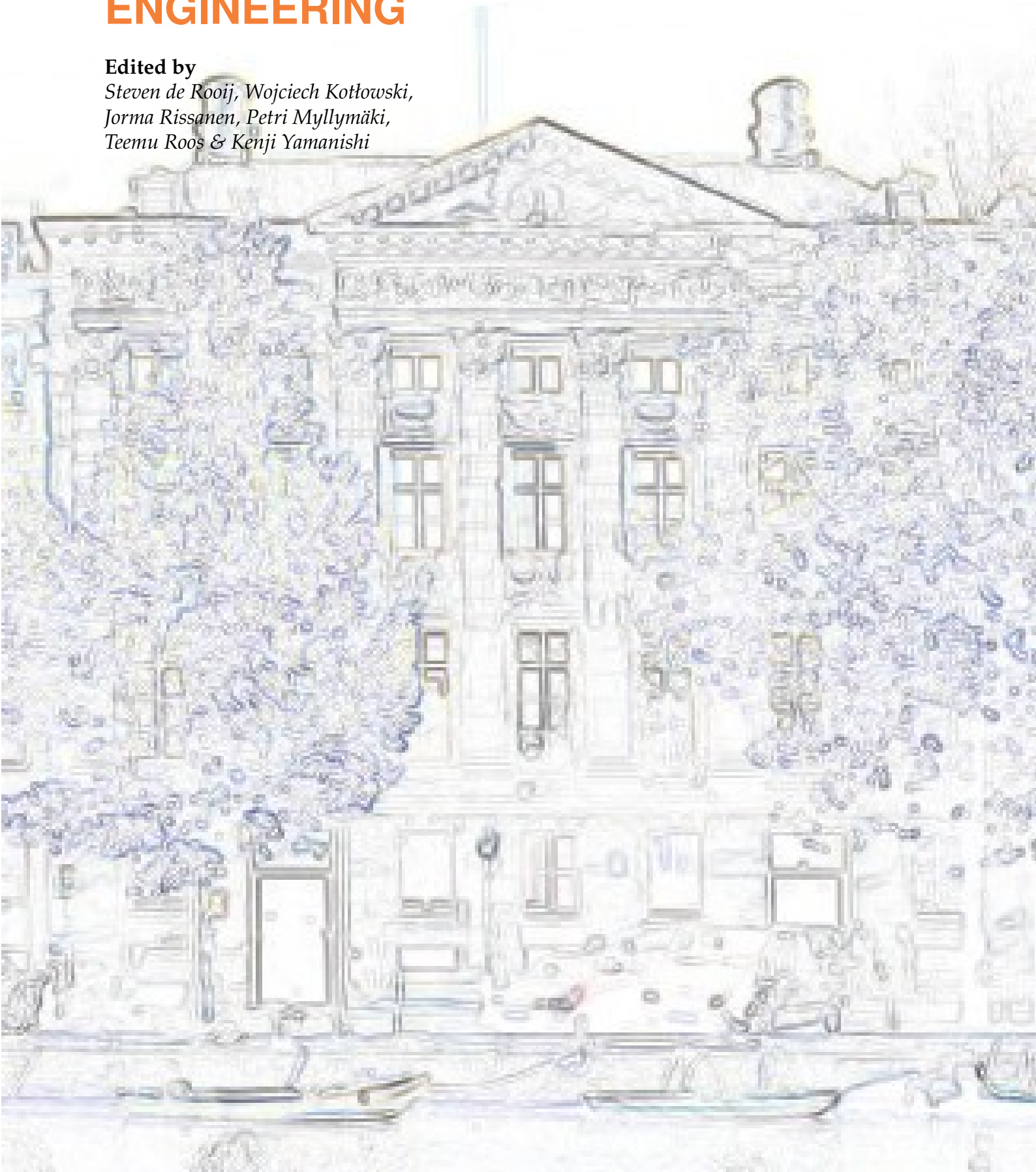


Proceedings of the  
**FIFTH WORKSHOP ON  
INFORMATION THEORETIC  
METHODS IN SCIENCE AND  
ENGINEERING**

**Edited by**

*Steven de Rooij, Wojciech Kottowski,  
Jorma Rissanen, Petri Myllymäki,  
Teemu Roos & Kenji Yamanishi*



# Preface

The Fifth Workshop on Information Theoretic Methods in Science and Engineering (WITMSE 2012) took place on August 27–30, 2012, in Amsterdam, Netherlands. This workshop series started in 2008. The first three iterations were hosted by the Technical University of Tampere, the fourth by the University of Helsinki and the Helsinki Institute for Information Technology HIIT, and this one by Centrum Wiskunde & Informatica (CWI), the national research institute for mathematics and computer science in Amsterdam, Netherlands. The event is sponsored by Stochastics – Theoretical and Applied Research (STAR).

As the title of the workshop suggests, WITMSE seeks speakers from a variety of disciplines with emphasis on both theory and applications of information and coding theory with special interest in modeling. Since the beginning our plan has been, and still is, to keep the number of participants small and to ensure the highest possible quality, which has been accomplished by inviting distinguished scholars as speakers.

This year's invitees include three plenary speakers: Peter Grünwald (CWI Amsterdam), Dominik Janzing (Max Planck Institute, Tübingen, Germany), and Rui M. Castro (Eindhoven University of Technology, Netherlands). Each has demonstrated a keen eye for the bigger picture, and is allotted a longer time slot to expand upon his views.

We would like to thank all the participants for their contributions to this event, and we hope that the extended abstracts that were submitted by many and that are collected in these proceedings will help make the ideas discussed at this workshop more easily accessible in the future.

September 3, 2012

Workshop chairs

<i>Steven de Rooij</i>	<i>Wojciech Kottowski</i>
<i>Jorma Rissanen</i>	<i>Kenji Yamanishi</i>
<i>Teemu Roos</i>	<i>Petri Myllymäki</i>

*Hosted by*



*Centrum Wiskunde & Informatica*

*Sponsored by*



*Stochastics - Theoretical and Applied Research*

# Contents

Preface . . . . .	1
<i>Thijs van Ommen</i> : Adapting AIC to conditional model selection . . . . .	4
<i>Boris Ryabko</i> : An information-theoretic method for estimating the performance of computer systems . . . . .	6
<i>Mervi Eerola and Satu Helske</i> : Analysing life history calendar data: a methodological comparison . . . . .	10
<i>Zhanna Reznikova and Boris Ryabko</i> : Application of information theory for studying numerical competence in animals: an insight from ants . . . . .	12
<i>Alberto Giovanni Busetto, Morteza Haghiri Chehreghani, and Joachim M. Buhmann</i> : Approximation set coding for information theoretic model validation . . . . .	14
<i>Daniil Ryabko</i> : Asymptotic statistics of stationary ergodic time series . . . . .	18
<i>Flemming Topsøe</i> : Beyond Shannon with three examples from geometry, statistics and information theory . . . . .	22
<i>Kenji Yamanishi, Ei-ichi Sakurai and Hiroki Kanazawa</i> : Change detection, hypothesis testing and data compression . . . . .	26
<i>So Hirai and Kenji Yamanishi</i> : Clustering change detection using Normalized Maximum Likelihood coding . . . . .	30
<i>Ralf Eggeling, Teemu Roos, Petri Myllymäki, and Ivo Grosse</i> : Comparison of NML and Bayesian scoring criteria for learning parsimonious Markov models . . . . .	32
<i>Kazuho Watanabe and Shiro Ikeda</i> : Convex Formulation for Nonparametric Estimation of Mixing Distribution . . . . .	36
<i>Guoqiang Zhang and Richard Heusdens</i> : Efficient message-passing for distributed quadratic optimization . . . . .	40
<i>Yuri Kalnishkan, Michael V. Vyugin, and Vladmimir Vovk</i> : Generalised entropies and asymptotic complexities of languages . . . . .	44
<i>Vladimir Vovk</i> : Informational and computational efficiency of set predictors . . . . .	48
<i>Hannes Wettig, Javad Nouri, Kirill Reshetnikov, and Roman Yangarber</i> : Information-theoretic methods for analysis and inference in etymology . . . . .	52
<i>Lukasz Dębowski</i> : Information-theoretic models of natural language . . . . .	56
<i>David Bickel</i> : Information-theoretic probability combination with applications to reconciling statistical methods . . . . .	59
<i>Anjali Mazumder and Steffen Lauritzen</i> : Information-theoretic value of evidence analysis using probabilistic expert systems . . . . .	63

<i>Ugo Vespier, Arno Knobbe, Siegfried Nijssen, and Joaquin Vanschoren: MDL-Based identification of relevant temporal scales in time series . . . . .</i>	64
<i>Jesús E. García, Verónica Andrea González-López, and M. L. L. Viola: Model selection for multivariate stochastic processes . . . . .</i>	68
<i>Erkki P. Liski and Antti Liski: Penalized least squares model averaging . . . . .</i>	72
<i>Wouter Koolen, Dimitri Adamskiy, and Manfred K. Warmuth: Putting Bayes to sleep . . . . .</i>	76
<i>Jesús E. García, Verónica Andrea González-López, and M. L. L. Viola: Robust model selection for stochastic processes . . . . .</i>	79
<i>Jilles Vreeken and Nikolaj Tatti: Summarising Event Sequences with Serial Episodes</i>	82
<i>Fares Hedayati and Peter L. Bartlett: The optimality of Jeffreys prior for online density estimation and the asymptotic normality of Maximum Likelihood estimators . . . . .</i>	86
 Author Index . . . . .	 90

# ADAPTING AIC TO CONDITIONAL MODEL SELECTION

*Thijs van Ommen*

Centrum Wiskunde & Informatica (CWI),  
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, Thijs.van.Ommen@cwi.nl

## ABSTRACT

In statistical settings such as regression and time series, we can condition on observed information when predicting the data of interest. For example, a regression model explains the dependent variables  $y_1, \dots, y_n$  in terms of the independent variables  $x_1, \dots, x_n$ . When we ask such a model to predict the value of  $y_{n+1}$  corresponding to some given value of  $x_{n+1}$ , that prediction's accuracy will vary with  $x_{n+1}$ . Existing methods for model selection do not take this variability into account, which often causes them to select inferior models.

One widely used method for model selection is AIC (Akaike's Information Criterion [1]), which is based on estimates of the KL divergence from the true distribution to each model. We propose an adaptation of AIC that takes the observed information into account when estimating the KL divergence, thereby getting rid of a bias in AIC's estimate.

## 1. A BIAS IN AIC

The principle underlying AIC and many subsequent criteria is that model selection methods should find the model  $g$  which minimizes

$$-2 E_{\mathbf{U}} E_{\mathbf{V}} \log g(\mathbf{V} | \hat{\theta}(\mathbf{U})), \quad (1)$$

where  $\hat{\theta}$  represents the maximum likelihood estimator in that model, and both random variables are independent samples of  $n$  data points each, both following the true distribution of the data. The inner expectation is the KL divergence from the true distribution to  $g(\cdot | \hat{\theta}(\mathbf{U}))$  up to a constant which is the same for all models. The quantity (1) can be seen as representing that we first estimate the model's parameters using a random sample  $\mathbf{U}$ , then judge the quality of this estimate by looking at its performance on an independent, identically distributed sample  $\mathbf{V}$ .

In regression, time series, and other settings, the data points consist of two parts  $u_i = (x_i, y_i)$ , and the models are sets of distributions on the *dependent variable*  $\mathbf{y}$  conditioned on the *independent variable*  $x$  (which may or may not be random). We call these *conditional* models. Then (1) can be adapted in two ways: as the extra-sample error

$$-2 E_{\mathbf{Y}|X} E_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' | X', \hat{\theta}(X, \mathbf{Y})), \quad (2)$$

and, replacing both  $X$  and  $X'$  by a single variable  $X$ , as the in-sample error

$$-2 E_{\mathbf{Y}|X} E_{\mathbf{Y}'|X} \log g(\mathbf{Y}' | X, \hat{\theta}(X, \mathbf{Y})). \quad (3)$$

The standard expression behind AIC (1) makes no reference to  $X$  or  $X'$ , which leads a straightforward derivation of AIC for a conditional model to make the tacit assumption  $X = X'$ , so that standard AIC estimates the in-sample error. This applies for instance to the well-known form of AIC for linear models, i.e. the residual sum of squares with a penalty of  $2k$ , where  $k$  is the model's order.

However, the extra-sample error (2) is more appropriate as a measure of the expected performance on new data. Using the in-sample error (3) instead results in a biased estimate of this performance. As the bias gets worse for larger models, this will lead to inferior model selection.

## 2. AN UNBIASED ADAPTATION

To get an estimator for (2), we do not make any assumptions about the process generating  $X$  and  $X'$  (it may not even be random) but treat their values as given. We denote the number of data points in  $X$  and  $X'$  by  $n$  and  $n'$ , respectively. In the case of simple linear regression with fixed variance, a derivation similar to AIC's leads to a penalty term of  $k + \kappa_{X'}$  in place of AIC's  $2k$ , where

$$\kappa_{X'} = \frac{n}{n'} \text{tr} \left[ X'^{\top} X' (X^{\top} X)^{-1} \right],$$

where  $X$  and  $X'$  represent design matrices. Similarly, a small sample corrected version analogous to AICc [2] can be derived and has penalty

$$k + \kappa_{X'} + \frac{(k + \kappa_{X'})(k + 1)}{n - k - 1}.$$

## 3. FOCUSED AIC FOR PREDICTION

If our goal is prediction, then the value  $X$  used in our derivation corresponds to the data we have observed already, and  $X'$  may be replaced by the single point  $x$  for which we need to predict the corresponding  $\mathbf{y}$ . This justifies treating  $X$  and  $X'$  as given in this practical setting. Thus we use  $x$  already at the stage of model selection, whereas standard methods for model selection only use it after selecting a model, to find the distribution of  $\mathbf{y}$  conditioned on that  $x$ . Then for the linear model with fixed

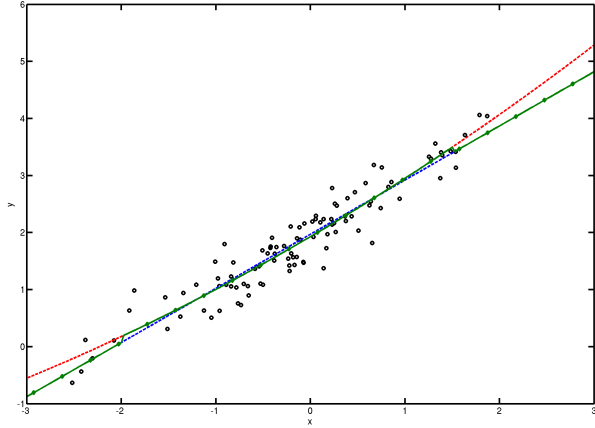


Figure 1. Example illustrating the result of applying FAIC to a sample of 100 data points. There are three models: the constant, linear, and quadratic functions; the true distribution uses a linear function. The choice of FAIC is marked in green: it selects a quadratic (red) function for  $x$  close to many observed data points, and a linear (blue) function elsewhere.

variance,  $\kappa_x$  becomes

$$\kappa_x = \frac{n}{n'} \text{tr}[xx^\top (X^\top X)^{-1}] = nx^\top (X^\top X)^{-1}x;$$

for unknown variance it becomes this value plus one.

We name this method Focused AIC. The term “focus” was first used by Claeskens and Hjort’s [3] to describe a model selection method that focuses on a parameter of interest when selecting a model. The behaviour of FAIC is illustrated in Figure 1.

#### 4. EXPERIMENTAL RESULTS

Simulation experiments with linear regression models indicate that our method outperforms AIC in terms of logarithmic (or squared) loss in many situations. Representative results are shown in Figure 2.

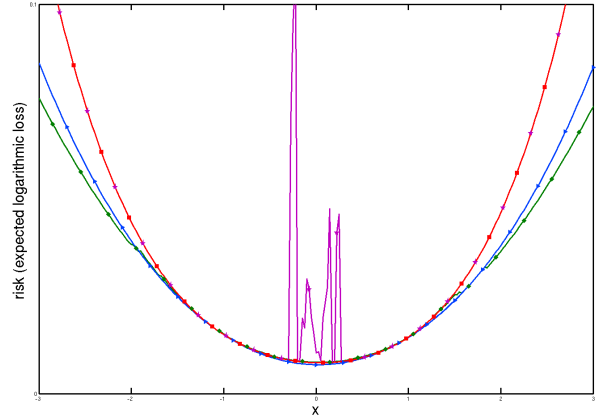


Figure 2. Average performance of different model selection methods as a function of  $x$ . Our FAIC (in green) outperforms the other methods for extreme  $x$  and is competitive otherwise; AIC (red) overfits especially for extreme  $x$ ; BIC (Bayesian Information Criterion, blue) is less likely to overfit than AIC; FIC (Focused Information Criterion, purple) is similar to AIC but selects a constant function in the center.

#### 5. REFERENCES

- [1] H. Akaike, “Information theory and an extension of the maximum likelihood principle,” in *Proc. 2nd International Symposium on Information Theory*, Tsahkadsor, Armenian SSR, Sep. 1971, pp. 267–281.
- [2] C. M. Hurvich and C-L. Tsai, “Regression and time series model selection in small samples,” *Biometrika*, vol. 76, pp. 297–307, Jun. 1989.
- [3] G. Claeskens and N. L. Hjort, “The focused information criterion,” *Journal of the American Statistical Association*, vol. 98, pp. 900–916, Dec. 2003.

# AN INFORMATION-THEORETIC METHOD FOR ESTIMATING THE PERFORMANCE OF COMPUTER SYSTEMS

*Boris Ryabko*

Siberian State University of Telecommunications and Information Sciences,  
Institute of Computational Technology of Siberian Branch of Russian  
Academy of Science, Novosibirsk, Russia; boris@ryabko.net

## ABSTRACT

We consider a notion of computer capacity as a novel approach to evaluation of computer performance. Computer capacity is based on the number of different tasks that can be executed in a given time. This characteristic does not depend on any particular task and is determined only by the computer architecture. It can be easily computed at the design stage and used for optimizing architectural decisions.

## 1. INTRODUCTION

The problem of computer performance evaluation attracts much research because various aspects of performance are the key goals of any new computer design, see, e.g., [1, 2]. Simple performance metrics, such as the number of integer or floating point operations executed per second, are not adequate for complex computer architectures we face today. A more appropriate and widely used approach is to measure performance by execution time of specially developed programs called benchmarks. The main issues of benchmarking are well known, we only mention a few. First, it is very difficult, if ever possible, to find an adequate set of tasks (in fact, any two different researchers suggest quite different benchmarks). Then, when a benchmark is used at the design stage, it must be run under a simulated environment which slows down the execution in many orders of magnitude, making it difficult to test various design decisions in the time-limited production process. As a consequence, the designers reduce the lengths and the number of benchmarks, which raises the question of conformity with real applications. Quite often, benchmarking is applied to already made devices for the purposes of evaluation and comparison. Here, the benchmarks produced by a hardware manufacturer may be suspected of being specially tuned just to facilitate sales. The benchmarks suggested by independent companies are prone to be outdated when applied to technologically novel devices. All these appeal to objectivity of evaluation results. The performance figures obtained in this way may be suitable for one kind of applications but useless for another.

We suggest a completely different approach to evaluation of computer performance which allows to circumvent the difficulties outlined above. The new approach is

based on calculation of the number of different tasks that can be executed in time  $T$ . This is quite similar to determining the channel capacity in information theory through the number of different signals that can be transmitted in a unit of time [3]. If one computer can execute, say,  $10^{10}$  different tasks in one hour while another one can execute  $10^{20}$  tasks, we may conclude that the latter computer is more capable in doing its work. The number of different tasks does not depend on any particular task and is determined only by the computer architecture which, in turn, is described by the instruction set, execution times of instructions, structure of pipelines and parallel processing units, memory structure and access time, and some other basic computer parameters. All these parameters can be set and adapted at the design stage to optimize the performance.

It is important to note that, generally speaking, the number of different tasks grows exponentially as a function of time. Indeed, if we have two different tasks  $X$  and  $Y$ , each executed in time  $T$ , then their succession  $XY$  will require  $2T$ , and the whole number of different tasks will grow from  $N$  to about  $N^2$  (not  $N^2$  exactly because there are some instructions that may start before and end after the moment  $T$ ). So we may write  $N(2T) \approx N^2(T)$  and, generally,  $N(kT) \approx N^k(T)$ , where  $N(T)$  denotes the number of task whose execution time equals  $T$ . This shows informally that the number of tasks grows exponentially as a function of time. Formal arguments will be presented below. So it makes sense to consider  $\log N(T)$  and to deal only with exponents, which may differ for different computers.

The idea of computer capacity was first suggested in [4, 5], where it was applied to Knuth's MMIX computer [7]. In this paper, we extend the approach to modern computers that incorporate cache memory, pipelines and parallel processing units. Thus we prepare a theoretical basis for determining capacities and making comparisons against benchmarks of well-known processors of Intel x86 family which was presented in [8].

## 2. COMPUTER CAPACITY

Denote by  $I = \{u_1, u_2, \dots, u_s\}$  the instruction set of a computer (processor). An admissible sequence of instructions  $X = x_1 x_2 \dots x_t, x_i \in I$ , seen as a process in time,

is called a computer task. The term “admissible” means that the instruction sequence  $X$  can be executed up to the last element without errors in computation (so-called exceptions), such as division by zero or illegal memory reference. We consider two tasks  $X$  and  $Y$  as different if they differ at least in one instruction, i.e., there is an  $i$  such that  $x_i \neq y_i$ . Notice also the difference between the computer task and the computer program. The task, as we think of it, is the flow of instructions executed by the processor. It is produced as a realization of some program. For example, if the program contains a loop which is to be iterated 100 times, the corresponding task will contain the body of the loop repeated 100 times.

Denote the execution time of instruction  $x$  by  $\tau(x)$ . Then the execution time  $\tau(X)$  of a task  $X$  is given by

$$\tau(X) = \sum_{i=1}^t \tau(x_i).$$

The number of different tasks whose execution time equals  $T$  may be written as

$$N(T) = |\{X : \tau(X) = T\}|.$$

The main performance characteristic which is essential in our approach, is the computer capacity  $C(I)$  defined as

$$C(I) = \limsup_{T \rightarrow \infty} \frac{\log N(T)}{T}. \quad (1)$$

Notice that this definition is virtually the same as the definition of channel capacity in [3], where  $N(T)$  means the number of different signal sequences of duration  $T$ . The majority of modern computers are synchronous devices, i.e., they operate in discrete time scale determined by a clock cycle. In this case  $\tau(x)$  can be measured in the number of processor cycles. It was shown in [5] that if all  $\tau(x)$  are integers with the greatest common divisor 1, then the limsup in (1) equals lim and always exists.

Notice also the following thing. Let there be given two computers with identical sets of instructions  $I_1$  and  $I_2$  apart that the first computer is twice faster than the second one, i.e.  $\tau_1(x) = \tau_2(x)/2$  for any  $x \in I_1$  ( $I_2$ ). From definition (1) we immediately obtain that the capacity of the first computer is two times greater than that of the second one, i.e.  $C(I_1) = 2C(I_2)$ . Apparently, this equation is quite natural.

The suggested approach can be applied to multiprocessor systems. Consider a computer system that consists of  $l$  processors which can operate independently. Let each  $j$ -th processor has an instruction set  $I_j$  and can perform  $N_j(T)$  tasks in time  $T$ . Then the total number of tasks  $N(T) = N_1(T)N_2(T) \cdots N_l(T)$ , and from (1) we have

$$C(\otimes_{j=1}^l I_j) = C(I_1) + C(I_2) + \dots + C(I_l), \quad (2)$$

where  $C(\otimes_{j=1}^l I_j)$  is the capacity of the considered multiprocessor system. In particular, the capacity of computer system with  $l$  identical processors is  $l$  times greater than the capacity of computer with one processor. The same arguments are relevant to distributed computer systems, or

computer networks. Note that (2) is not a simple sum if the processors have some shared resources, such as shared memory. In this case the individual capacities must be diminished due to competitions for shared resources.

The definition of computer capacity is quite general, it does not restrain us from using one or other model of computer task formation. We may apply restrictions on instruction sequences, consider dependence of instruction execution times upon preceding instructions, and so on. Generally, the calculation of the limit in (1) becomes a complicated combinatorial problem. But as a first step, we can use a simple method suggested by Shannon in [3] for finding the capacity of noiseless channel where code symbols had different durations. When we use this simple method, we assume that all sequences of instructions are admissible. Clearly, by doing that we obtain an upper bound of capacity, which we denote by  $\hat{C}(I)$ , because the number of admissible instruction sequences  $N(T)$  cannot be larger than the number of all possible sequences, denoted thus by  $\hat{N}(T)$ . Despite this simplification, we take proper account of the effects of caches, pipelines and parallel processing, as will be shown below. More specifically, following [3], for the instruction set  $I = \{u_1, u_2, \dots, u_s\}$  we may state that the number of all possible instruction sequences must satisfy the difference equation

$$\hat{N}(T) = \hat{N}(T - \tau_1) + \hat{N}(T - \tau_2) + \dots + \hat{N}(T - \tau_s).$$

Here  $\hat{N}(T - \tau_j)$  is the number of instruction sequences of duration  $T$  ending in instruction  $u_j$ . It is well-known from the theory of finite differences that asymptotically, as  $T \rightarrow \infty$ ,  $\hat{N}(T) = Z_0^T$ , where  $Z_0$  is the greatest positive root of the characteristic equation

$$Z^{-\tau(u_1)} + Z^{-\tau(u_2)} + \dots + Z^{-\tau(u_s)} = 1. \quad (3)$$

So from the definition of computer capacity (1) we have

$$\hat{C}(I) = \log Z_0.$$

In what follows we will estimate  $\hat{C}(I)$  as a first approximation of real computer capacity, realizing that there are more complicated and more exact methods of finding  $C(I)$ .

Consider some examples. Let the first computer has only two instructions and execution time of each instruction is one clock cycle. So we have  $I_1 = \{u_1, u_2\}$ ,  $\tau(u_1) = \tau(u_2) = 1$  and the characteristic equation is  $2Z^{-1} = 1$ . Hence  $Z_0 = 2$  and the computer capacity  $C(I_1) = \log 2 = 1$  bit per cycle. Now add a third instruction with duration 2 cycles:  $I_2 = \{u_1, u_2, u_3\}$ ,  $\tau(u_1) = \tau(u_2) = 1$ ,  $\tau(u_3) = 2$ . The characteristic equation is  $2Z^{-1} + Z^{-2} = 1$ , its greatest root  $Z_0 = 2.414$ . The capacity  $C(I_2) = 1.27$  bit per cycle, it is greater than  $C(I_1)$  due to “more rich” instruction set  $I_2$ .

In practice, the computer instructions are often built of operation codes and operands, which may be references to internal registers, memory, or some immediate data. The key point is that to find the computer capacity we must consider the instruction set containing all operations with



all combinations of operands. Let, for example, the computer have 8 registers,  $2^{16}$  memory locations, and can perform two operations op1 and op2 of the following format: (op1 reg reg) and (op2 reg mem), where reg is one of 8 registers, and mem is a reference to one of  $2^{16}$  memory locations. Let op1 require 1 cycle and op2 2 cycles. Then the characteristic equation will be

$$\frac{8 \cdot 8}{Z} + \frac{8 \cdot 2^{16}}{Z^2} = 1.$$

The solution  $Z_0 = 757$  and  $C(I_3) = 9.56$  bits per cycle.

### 3. ENTROPY EFFICIENCY

It should be noted that to calculate the computer capacity, no probabilities or frequencies of instructions are needed. It does not mean that all the instructions are assumed to be equiprobable. In fact, the capacity is attained if the instructions appear with some “optimal” probabilities. In other words, the capacity is a maximal value which can be obtained if we use the processor instructions with certain frequencies. A connection between the computer capacity and various probabilistic models is established with the aid of the notion of entropy efficiency. There the sense of “optimal” probabilities mentioned above is clarified.

Consider the situation when computer is used for solving a particular kind of problems. For example, we use computer for solving differential equations. In this case the set of tasks to be performed is a subset of all possible tasks. We assume that the tasks of the set of interest can be modeled as realizations of a stationary and ergodic stochastic process. Let  $X = x_1 x_2 x_3 \dots$  be a sequence of random variables taking values over instruction set  $I$ . Denote by  $P_X(w)$  the probability that  $x_1 x_2 \dots x_{n+1} = w$ ,  $w \in I^{n+1}$  for any  $n \geq 0$ . The entropy rate is defined as usually, see, e.g., [9]:

$$h(X) = \lim_{n \rightarrow \infty} -\frac{1}{n+1} \sum_{w \in I^{n+1}} P_X(w) \log P_X(w).$$

Now the entropy efficiency, as a measure of computer performance, is defined as follows:

$$c(I, X) = h(X) / \sum_{u \in I} P_X(u) \tau(u). \quad (4)$$

In other words,  $c(I, X)$  is the ratio of the entropy rate of instruction flow  $X$  to the average execution time of instruction.

To motivate this definition, notice that if we take a large integer  $t$  and consider all  $t$ -element instruction sequences  $x_1 \dots x_t$ , then the number of “typical” sequences will be approximately  $2^{th(X)}$ , whereas the total execution time of any sequence will be approximately  $t \sum_{u \in I} P_X(u) \tau(u)$ . (By definition of a typical sequence, the frequency of any word  $w$  in it is close to the probability  $P_X(w)$ . The total probability of the set of all typical sequences is close to 1.) So the ratio between  $\log(2^{th(X)})$  and the average execution time will be asymptotically equal to (4) if  $t \rightarrow \infty$ .

This observation shows the relation between computer capacity (1) and entropy efficiency: the former is defined through the number of all tasks, the latter through the number of typical tasks, executed in one time unit. Another conclusion from this consideration is that

$$c(I, X) \leq C(I). \quad (5)$$

Now we shall say some words about estimation of the entropy efficiency. To do that we must observe the flow of instructions generated by the application of interest. Then we may use any method known in Information Theory to estimate the entropy of the instruction sequence and probabilities of particular instructions. Again, the simplest approach is to consider the case where all instructions are independent and identically distributed (i.i.d. sequence). In this situation the definition of entropy efficiency may be re-written in the following form:

$$\hat{c}(I, X) = - \sum_{u \in I} P_X(u) \log P_X(u) / \sum_{u \in I} P_X(u) \tau(u).$$

It can be easily checked now by direct calculation that if  $P_X(u) = Z_0^{-\tau(u)}$  for all  $u \in I$ , where  $Z_0$  is the greatest root of characteristic equation (3), then

$$\hat{c}(I, X) = \log Z_0 = \hat{C}(I),$$

i.e. the entropy efficiency reaches the computer capacity and is maximal according to (5).

### 4. COMPUTER CAPACITY IN MODERN COMPUTER ARCHITECTURES

The most essential elements of modern computer architectures that influence the capacity defined in (1) are cache memory (usually organized in several levels), parallel execution units (such as floating point unit), instruction pipelines and closely connected branch predictors, and multiple cores (including such technologies as hyperthreading). In this section, we address all these issues and show simple ways of their solution when determining computer capacity.

To assess the effect of cache memory on computer capacity we observe what happens at every time instant. Let, for example, instruction “ADD REG, MEM” is executed which adds a word in memory to a register and stores the result in the register. In our approach to estimation of capacity we assume that any register and any memory location can be accessed. Let there be  $R$  registers and  $M$  words in memory available. To show the main idea, consider a cache memory consisting of two levels L1 and L2 of sizes  $L_1$  and  $L_2$ , respectively. If the address MEM hits L1 cache, let the execution time of the instruction be  $\tau_{L1}$ . Otherwise, if the address hits L2 cache, let the execution time be  $\tau_{L2}$  (usually, much greater than  $\tau_{L1}$ ). If the address is not cached, let the execution time be  $\tau_M$  (usually, much more greater than  $\tau_{L2}$ ). Suppose that L1 and L2 are not exclusive, i.e. a memory location cached in L1 is also cached in L2. Then the corresponding part of characteristic equation will look like this:

$$RL_1 \frac{1}{Z^{\tau_{L1}}} + R(L_2 - L_1) \frac{1}{Z^{\tau_{L2}}} + R(M - L_2 - L_1) \frac{1}{Z^{\tau_M}}.$$

If  $L_1$  and  $L_2$  are exclusive then we should not subtract  $L_1$  in the second summand. All other processor instructions that operate with memory can be considered similarly.

The other issue is the presence of some units  $U_1, U_2, \dots$  that can operate concurrently with the “main” part that performs basic operations (e.g., FPU, MMX and XMM blocks in x86 processors). Although the instructions executed by those units usually alternate with basic instructions and may have dependences, to find an upper bound on computer capacity we may consider these units as independent processors, i.e. to find their own capacities and sum them up according to (2). However, we must take into account that some units may be mutually exclusive (e.g., FPU and MMX blocks cannot operate concurrently in x86 processors since they are based on one and the same register pool [10]). The solution is to consider all subsets of mutually compatible units and calculate capacities of those subsets. Then, since we are interested in an upper bound of computer capacity, we may choose the greatest capacity estimate. For example, there are two compatible subsets in x86 processors: MAIN + FPU + XMM and MAIN + MMX + XMM (obviously, there is no need to consider the subsets of smaller sizes). The subset having greater capacity determines the capacity of the whole computer.

The next architectural feature is the pipeline processing combined with branch prediction. Instruction timings provided in documentation assume that the pipeline is optimally filled, i.e., there are no empty stages and execution time is determined solely by the complexity of instruction. However, the pipeline operation is stopped when a mispredicted branch occurs. The instruction that must follow the mispredicted branch is delayed for the number of cycles,  $k$ , equal to the number of pipeline stages from the fetch stage to the execute stage. The next instruction is delayed for  $k - 1$  cycles and so on. The exact model would require to consider all  $k$ -element instruction sequences with any mispredicted branch. But we prefer a simpler way, sufficient for obtaining an upper bound of capacity. Assume that after any mispredicted branch we wait for  $k$  cycles before the execution of next instructions. That is, the execution time of mispredicted jump instruction is increased by  $k$  cycles. Since the computer capacity is defined through the number of all computer tasks, we can separately consider predicted and mispredicted jump instructions.

Finally, we address the problem of parallelism which is essential in hyper-threading and multicore technologies.

It is demonstrated there that computer performance indicators obtained through calculation of computer capacity and by benchmarks are very close to each other. So the computer capacity approach definitely can be used at the design stage when benchmarking is time-consuming or not at all possible.

## 5. REFERENCES

[1] W. Stallings, *Computer Organization and Architecture: Designing for Performance*. Prentice-Hall, 2009.

- [2] A. S. Tanenbaum, *Structured Computer Organization*. Prentice Hall, 2005.
- [3] C. E. Shannon, “A mathematical theory of communication,” *Bell Sys. Tech. J.*, Vol. 27, 1948, pp. 379–423, pp. 623–656.
- [4] B. Ryabko, “Using information theory to study the efficiency and capacity of computers and similar devices,” *Proc. of the 2010 Workshop on Information Theoretic Methods in Science and Engineering (Tampere, Finland, 16-18 August 2010)* .
- [5] B. Ryabko , “On the efficiency and capacity of computers,” *Applied Mathematics Letters*, v. 25, 2012, pp. 398 - 400
- [6] B. Ryabko, “An information-theoretic approach to estimate the capacity of processing units,” *Performance Evaluation*, V. 69, 2012, pp. 267–273.
- [7] D. E. Knuth, *The Art of Computer Programming. Vol. 1, Fascicle 1: MMIX – A RISC Computer for the New Millennium*. Addison-Wesley, 2005.
- [8] A. Fionov, Yu. Polyakov, and B. Ryabko, “Application of computer capacity to evaluation of Intel x86 processors,” *2nd International Congress on Computer Applications and Computational Science*, November 15–17, 2011, Bali, Indonesia, (Springer, *Advances in Intelligent and Soft Computing*, Vol. 145, 2012, pp. 99–104).
- [9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006.
- [10] Intel 64 and IA-32 Architectures Software Developers Manual Volume 1: Basic Architecture, Intel Corp., 2011.

# ANALYSING LIFE HISTORY CALENDAR DATA: A METHODOLOGICAL COMPARISON

*Mervi Eerola*<sup>1</sup>, *Satu Helske*<sup>2</sup>

<sup>1</sup> Department of Mathematics and Statistics,

FIN-20014 University of Turku, FINLAND, mervi.eerola@utu.fi,

<sup>2</sup> Methodology Centre for Human Sciences/Department of Mathematics and Statistics,  
P.O.Box 35, FIN-40014 University of Jyväskylä, FINLAND, satu.helske@jyu.fi

## ABSTRACT

The life history calendar, also called an event-history calendar, is a data-collection tool for obtaining reliable retrospective data about life events. The advantage of a life history calendar is that the order and proximity of important transitions in multiple life domains can be studied at the same time.

To illustrate the analysis of such data, we compare the model-based probabilistic event history analysis and a more recent type of approach of model-free data-mining, sequence analysis. The latter is well known in bioinformatics in the analysis of protein or DNA sequences. In life course analysis it is less familiar but has provided novel insight to the diversity of life trajectories and their relationship to life satisfaction. We emphasize the differences, but also the complementary advantages of the methods.

In event history analysis, we consider the data generated by a marked point process  $(T_n, X_n)_{n \geq 1}$ , a time-ordered sequence of points or events, characterised by pairs of random variables, the occurrence times  $T_1, T_2, \dots$  and marks  $X_1, X_2, \dots$  describing what happens at a particular  $T$ . Instead of transition hazards, we estimate the cumulative prediction probabilities of a particular life event in the entire observed trajectory, given the history of the marked point process. This way of combining information in multi-state event history models has been called 'survival synthesis'. The innovation gain from

observing a life event at a particular age, related to the prediction of another life event, can be quantified and monitored visually.

In sequence analysis, we compare several dissimilarity measures between the life sequences, either assuming independence or using some *ad hoc* definition of dependence between the sequence elements. We also contrast data-driven (estimated) and user-defined costs of substituting one sequence element with another.

As an example, we study young adults' transition to adulthood as a sequence of events in three life domains (partnership, parenthood and employment). The events define the multi-state event history model and the parallel life domains in the multidimensional sequence analysis.

We conclude that the two approaches complement each other in life course analysis; sequence analysis can effectively find typical and atypical life patterns while event history analysis is needed for causal inquiries.

**Keywords:** Distance-based data; Life course analysis, Life history calendar; Multidimensional sequence analysis; Multi-state model; Prediction probability

## 1. REFERENCES

- [1] Avshalom Caspi, Terrie E. Moffitt, Arland Thornton, Deborah Freedman, et al., "The life history calendar: A research and clinical assessment method for collecting retrospec-

tive event-history data.” *International Journal of Methods in Psychiatric Research*, vol. 6, no. 2, pp. 101–114, 1996.

- [2] M. Eerola, *Probabilistic causality in longitudinal studies*, vol. 92 of *Lecture Notes in Statistics*, Springer-Verlag, 1994.
- [3] Jacques-Antoine Gauthier, Eric D. Widmer, Philipp Bucher, and Cédric Notredame, “Multichannel sequence analysis applied to social science data,” *Sociological Methodology*, vol. 40, no. 1, pp. 1–38, 2010.
- [4] I. Tabus J. Helske, M. Eerola, “Minimum description length based hidden markov model clustering for life sequence analysis,” in *Proc. Third WITMSE Conf.*, Tampere, Finland, Aug. 2010.
- [5] G. Pollock, “Holistic trajectories: a study of combined employment, housing and family careers by using multiple-sequence analysis,” *J. R. Statist. Soc. A*, vol. 170, pp. 167–183, Dec. 2007.

# APPLICATION OF INFORMATION THEORY FOR STUDYING NUMERICAL COMPETENCE IN ANIMALS: AN INSIGHT FROM ANTS

Zhanna Reznikova<sup>1</sup> and Boris Ryabko<sup>2</sup>

<sup>1</sup> Institute of Systematics and Ecology of Animals, Siberian Branch RAS, Frunze 11, Novosibirsk, 630091, and Novosibirsk State University, Novosibirsk, RUSSIA, zhanna@reznikova.net

<sup>2</sup> Siberian State University of Telecommunication and Computer Science, and Institute of Computing Technologies, Siberian Branch RAS, Novosibirsk, RUSSIA, boris@ryabko.net

## ABSTRACT

Our long – term experimental study on ant “language” and intelligence fully based on ideas of information theory revealed a symbolic language in highly social ant species and demonstrated these insects as being able to transfer to each other the information about the number of objects and can even add and subtract small numbers in order to optimize their messages. We suggest that application of ideas of information theory can open new horizons for studying numerical competence in non-human animals.

## 1. INTRODUCTION

Since C. Shannon [1] published his influential paper “A mathematical theory of communication”, the fundamental role of information theory has been appreciated not only in its direct applications, but also in robotics, linguistics and biology. Numerical competence is one of the main intriguing domains of animal intelligence. Recent studies have demonstrated some species, from mealy beetles to elephants, as being able to judge about numbers of stimuli, including things, and sounds, and even smells (see [2] for a review); however, we are still lacking an adequate “language” for comparative analysis. The main difficulty in comparing numerical abilities in humans and other species is that our numerical competence is closely connected with abilities for language usage and for symbolic representation. We suggested a new experimental paradigm which is based on ideas of information theory and is the first one to exploit natural communicative systems of animals [3]. Ants of highly social species are good candidates for studying general rules of cognitive communication. There are more than 12000 ant species on Earth, and the great majority of them use relatively simple forms of communication such as odour trails, tandem running, and so on. Only a few highly social species belong to the elite club of rare “cognitive specialists”, and among them are several species of red wood ants (*Formica rufa* group), with their big anthills

“boiling” with hundreds of thousands of active individuals [4].

## 2. IDEAS, METHODS AND RESULTS

In our experiments scouts of red wood ants were required to transfer to foragers in a laboratory nest the information about which branch of a special “counting maze” they had to go to in order to obtain syrup. The main idea of this experimental paradigm is that experimenters can judge how ants represent numbers by estimating how much time individual ants spend on “pronouncing” numbers, that is, on transferring information about index numbers of branches, that is, the information about which branch of a special “counting maze” they had to go to in order to obtain syrup. The main idea is that experimenters can judge how ants represent numbers by estimating how much time individual ants spend on “pronouncing” numbers, that is, on transferring information about index numbers of branches. The findings concerning number-related skills in ants are based on comparisons of duration of information contacts between scouts and foragers which preceded successful trips by the foraging teams.

It turned out that the relation between the index number of the branch ( $j$ ) and the duration of the contact between the scout and the foragers ( $t$ ) is well described by the equation

$$t = c j + d$$

for different set-ups which are characterized by different shapes, distances between the branches and lengths of the branches. The values of parameters  $c$  and  $d$  are close and do not depend either on the lengths of the branches or on other parameters.

It is interesting that quantitative characteristics of the ants’ “number system” seem to be close, at least outwardly, to some archaic human languages: the length of the code of a given number is proportional to its value. For example, the word “finger” corresponds to 1, “finger, finger” to the number 2, “finger, finger, finger” to the number 3 and so on. In modern human languages the length of the code word of a number  $j$  is

approximately proportional to  $\log j$  (for large  $j$ 's), and the modern numeration system is the result of a long and complicated development.

An experimental scheme for studying ants' "arithmetic" skills based on a fundamental idea of information theory, which is that in a "reasonable" communication system the frequency of usage of a message and its length must correlate. The informal pattern is quite simple: the more frequently a message is used in a language, the shorter is the word or the phrase coding it. This phenomenon is manifested in all known human languages

The scheme was as follows. Ants were offered a horizontal trunk with 30 branches. The experiments were divided into three stages, and at each of them the regularity of placing the trough with syrup on branches with different numbers was changed. At the first stage, the branch containing the trough with syrup was selected randomly, with equal probabilities for all branches. So the probability of the trough with syrup being placed on a particular branch was  $1/30$ . At the second stage we chose two "special" branches A and B (N 7 and N 14; N 10 and N 20; and N 10 and N 19 in different years) on which the trough with syrup occurred during the experiments much more frequently than on the rest - with a probability of  $1/3$  for "A" and "B", and  $1/84$  for each of the other 28 branches. In this way, two "messages" - "the trough is on branch A" and "the trough is on branch B" - had a much higher probability than the remaining 28 messages. In one series of trials we used only one "special" point A (the branch N 15). On this branch the food appeared with the probability of  $1/2$ , and  $1/58$  for each of the other 29 branches. At the third stage of the experiment, the number of the branch with the trough was chosen at random again.

The obtained data demonstrated that ants appeared to be forced to develop a new code in order to optimize their messages, and the usage of this new code has to be based on simple arithmetic operations. The patterns of dependence of the information transmission time on the number of the food-containing branch at the first and third stages of experiments were considerably different. In the vicinities of the "special" branches, the time taken for transmission of the information about the number of the branch with the trough was, on the average, shorter.

For example, in the first series, at the first stage of the experiments the ants took 70–82 seconds to transmit the information about the fact that the trough with syrup was on branch N 11, and 8–12 seconds to transmit the information about branch N 1. At the third stage it took 5–15 seconds to transmit the information about branch N 11.

Analysis of the time duration of information transmission by the ants raises the possibility that at the third stage of the experiment the scouts' messages consisted of two parts: the information about which of the "special" branches was the nearest to the branch with the trough, and the information about how many branches away is the branch with the trough from a certain "special" branch. In other words, the ants,

presumably, passed the "name" of the "special" branch nearest to the branch with the trough, and then the number which had to be added or subtracted in order to find the branch with the trough.

That ant teams went directly to the "correct" branch enables us to suggest that they performed correctly whatever "mental" operation (subtraction or addition) was to be made.

It is likely that at the third stage of the experiment the ants used simple additions and subtractions, achieving economy in a manner reminiscent of the Roman numeral system when the numbers 10 and 20, 10 and 19 in different series of the experiments, played a role similar to that of the Roman numbers V and X. This also indicates that these insects have a communication system with a great degree of flexibility. Until the frequencies with which the food was placed on different branches started exhibiting regularities, the ants were "encoding" each number ( $j$ ) of a branch with a message of length proportional to  $j$ , which suggests unitary coding. Subsequent changes of code in response to special regularities in the frequencies are in line with a basic information-theoretic principle that in an efficient communication system the frequency of use of a message and the length of that message are related.

The obtained results show that information theory is not only excellent mathematical theory, but many of its results may be considered as Nature laws.

### 3. REFERENCES

- [1] Shannon C.E. "A Mathematical Theory of Communication", Bell System Technical Journal, Vol. 27, pp.379–423, 623–656, 1948.
- [2] Reznikova Z. *Animal Intelligence: From Individual to Social Cognition*. Cambridge University Press, 2007.
- [3] Reznikova Z., Ryabko B. "Numerical competence in animals, with an insight from ants", *Behaviour*, Volume 148, Number 4, pp. 405-434, 2011.
- [4] Reznikova Z. "Experimental paradigms for studying cognition and communication in ants (Hymenoptera: Formicidae)", *Myrmecological News*, Volume 11, pp. 201 – 214.

# APPROXIMATION SET CODING FOR INFORMATION THEORETIC MODEL VALIDATION

Alberto Giovanni Busetto<sup>1,2</sup>, Morteza Haghir Chehreghani<sup>1</sup>, Joachim M. Buhmann<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, ETH Zurich, Zurich, Switzerland

<sup>2</sup>Competence Center for Systems Biology and Metabolic Diseases, Zurich, Switzerland

## ABSTRACT

Models can be seen as mathematical tools aimed at prediction. The fundamental modeling question is: which model best generalizes the available data? We discuss the central ideas of a recently introduced principle for model validation: Approximation Set Coding (ASC). The principle is inspired by concepts from statistical physics and it is based on information theory. There exists a central analogy between communication and learning which can be used to evaluate informativeness by designing codes based on sets of solutions. These sets are called approximation sets; they should be small enough to be informative and large enough to be stable under noise fluctuations. We present the application of ASC to two tasks: clustering and learning of logical propositions. The two modeling tasks highlight the generality of the principle and its main properties. Experimental results are discussed in the biological application domain.

## 1. INTRODUCTION

In the context of modeling, validation constitutes a fundamental step. The central question is: which model should be selected given the data? A justified answer to this question requires a precise assessment of the predictive capability of candidate models.

Our problem definition explicitly considers the case in which models are defined in terms of cost functions. This setting is in contrast to the more restrictive (yet still interesting) one in which a specific cost is given *a priori* and the estimation process solely consists of selecting the best parameters from a set. In our case, model selection consists of finding the most informative cost. To do that, we must define and estimate informativeness.

Let us start by introducing cluster model selection as a motivating example. We define a solution of a clustering analysis as an assignment of labels to samples. Clustering, hence, produces partitions of the available sample points. Alternative partitions are evaluated and selected on the basis of a cost function. The cost function (that is, the model) is often made explicit, but may also be implicitly defined in terms of outputs of an algorithmic process. In applications, the cost function is typically chosen according to human intuition and remains fixed for the analysis. For simplicity, let us now consider a clustering procedure based on an explicit cost function  $R(\cdot|X)$ , which evalu-

ates solutions on the basis of the dataset  $X$ . Given  $X$ , the learning process terminates as soon as a (globally or locally) optimal solution is found. At this point, two important issues remain open. Is the result informative? Is the model justified? In order to answer these questions, we need a precise definition of the modeling goal in terms of predictive capabilities. There already exist theoretical and practical answers to these questions. At present, the set of established principles and procedures for predictive modeling include Minimum Description Length [1], Kolmogorov Structure Function [2], BIC [3] & AIC [4], Minimum Message Length [5], Solomonoff's Induction [6, 7], PAC [8] and PAC-Bayesian generalization bounds [9]. These approaches are based on convincing justifications from information theory, algorithmic information theory, probability and statistical learning theory.

The discussion of the individual merits of these approaches is certainly of great interest and value but goes beyond the scope of this contribution. We focus on the recently introduced idea of Approximation Set Coding [10]. ASC shares the spirit of the mentioned approaches, but with a rather different goal: selecting models by measuring the informativeness of equivalence classes of solutions.

## 2. APPROXIMATION SET CODING

ASC selects the optimal quantization of the hypothesis class to find the set of hypotheses constituting the best tradeoff between informativeness and stability. The informal justification is the following. On the one hand, selecting very few solutions exposes the modeler to the danger of instability with respect to fluctuations induced by noise [11]. On the other hand, selecting many solutions yields stable but rather uninformative results. With minimalistic assumptions about the nature of the noise, it is possible to select the set of solutions which provides the best tradeoff between informativeness and stability. This optimal set constitutes the best approximation available for a model. Models are then compared in terms of their informativeness, finally yielding the optimal approximation set.

Let us now start by formalizing the central concepts. Consider a cost model  $R(c|X)$ , which evaluates the cost of choosing solution  $c \in \mathcal{C}(X)$  to generalize the given dataset  $X \in \mathcal{X}$ . As conventional in statistical learning

theory, the smaller the cost, the better is the quality of the solution. The set of all candidate solutions is defined as the hypothesis class  $\mathcal{C}(X)$ , which is given to the modeler. Depending on the application, individual solutions might be parametric (with variable parameters) or simple elements from a set. In both cases, each element  $c$  of the hypothesis class indicates a particular and fixed candidate solution. Different cost functions define different models (for instance  $R_1(c|X)$  and  $R_2(c|X)$ ); for the rest of the manuscript, we identify models with their respective cost function. Our task is then to evaluate a set of models and select the best one, that is the most predictive. For each cost model  $R(\cdot|X)$  and a given dataset, the optimal solutions are provided by the set of empirical minimizers

$$\mathcal{C}^\perp(X) = \arg \min_{c \in \mathcal{C}(X)} R(c|X). \quad (1)$$

Since costs are evaluated as a function of the data, we must take into account the variability with respect to  $X$ . In order to perform this step, we consider the minimal case in which two datasets (each of size  $n$ ) are available to the modeler. The extension to settings with a larger number of sample sets is straightforward and exhibits analogous results. We assume that two datasets  $X_1$  and  $X_2$  are drawn independently from the same distribution. Since the hypothesis class might also depend on the dataset, we need a way to map solutions from  $\mathcal{C}(X^1)$  to  $\mathcal{C}(X^2)$ . Transferring solutions between instances is a necessary requirement to evaluate the generalization properties from training to test data. For that, we introduce the mapping function  $\psi : \mathcal{C}(X^1) \rightarrow \mathcal{C}(X^2)$ .

By mapping the solutions from one dataset to another,  $\psi$  allows the modeler to map solutions across instances (for instance, by mapping to the nearest neighbor). For every subset of solutions  $A \subseteq \mathcal{C}(X_1)$ , we denote the mapped subset as

$$\psi \circ A = \{\psi(a), a \in A\} \subseteq \mathcal{C}(X_2). \quad (2)$$

In case of noise, the set of mapped empirical minimizers do not necessarily coincide with the solutions induced by the second dataset. The intersection  $\psi \circ \mathcal{C}^\perp(X_1) \cap \mathcal{C}^\perp(X_2)$  might be small or even empty. In fact, fluctuations in the data might induce perturbations in the empirical minimizers, which will tend to diverge from each other as the noise level increases. Instead of taking the two sets of empirical minimizers (to avoid inconsistency due to instability), we consider larger sets of solutions. These sets are called approximation sets and are defined as a function of a parameter  $\gamma$  so that

$$\mathcal{C}_\gamma(X_i) = \{c \in \mathcal{C}(X_i) : R(c|X_i) \leq R_\perp(X_i) + \gamma\} \quad (3)$$

for  $i = 1, 2$ . These sets are  $\gamma$ -close to the solution costs  $R^\perp(X_i) := R(c_i^\perp|X_i)$  of the respective empirical minimizers  $c_i^\perp \in \mathcal{C}^\perp(X_i)$ ,  $i = 1, 2$ . At this point, the question is which  $\gamma$  should we select? For  $\gamma = 0$  we get only the empirical minimizers. If  $\gamma$  is too small, the results are unstable. For too large  $\gamma$ , the selection tends to include all the entire hypothesis class (thus yielding uninformative

results). The communication analogy is introduced to address this question. It is based on the sender-receiver scenario in which distinguishing individual solutions based on data corresponds to transmitting messages over a noisy channel. The communication capacity reflects the ability to discriminate solutions through the applied transformations. Ultimately, the success of the communication depends on noise level and coding strategy.

The communication process for a certain  $\gamma$  is described by the following procedures:

- *Coding:*

1. Sender and receiver agree on  $R$  and share  $X_1$ .
2. They both calculate the  $\gamma$ -approximation sets.
3. The sender generates a set of transformations  $\Sigma = \{\sigma : \mathcal{X} \rightarrow \mathcal{X}\}$  which define a set of training optimization problems  $R(\cdot|\sigma \circ X_1)$  and their respective  $\gamma$ -approximation sets.
4. The sender sends  $\Sigma$  to the receiver which calculates the approximation sets for each transformation.

- *Transmission:*

1. The sender is a stationary source: it selects a transformation  $\sigma_s$  as message without directly revealing it to the receiver.
2. The transformation  $\sigma_s$  is applied by the sender to  $X_2$ .
3. The transformed dataset  $\sigma_s \circ X_2$  is sent to the receiver.
4. The receiver has to reconstruct the transformation  $\sigma_s$  from the approximation set of  $\sigma \circ X_2$  without directly knowing  $X_2$  and  $\sigma_s$ .

Each transformation  $\sigma_s$  generated by the sender is estimated by the receiver through the decoding rule

$$\hat{\sigma} = \arg \max_{\sigma \in \Sigma} |\psi \circ \mathcal{C}_\gamma(\sigma \circ X_1) \cap \mathcal{C}_\gamma(\sigma_s \circ X_2)|. \quad (4)$$

Decoding is possible because, in contrast to  $\sigma_s$  and  $X_2$ ,  $\sigma_s \circ X_2$  is known to the receiver. It can be used to calculate the approximation sets used to uniquely identify  $\sigma_s$ . The aim is the following: achieving optimal communication (which is reliable and informative). Approximation sets define codebook vectors; while large  $\gamma$  correspond to small sets of distinct vectors for coding, small  $\gamma$  might correspond to higher error rates for decoding.

Communication errors are due to wrong decoding, that is when  $\hat{\sigma} \neq \sigma_s$ . The probability of a communication error is hence given by

$$P(\hat{\sigma} \neq \sigma_s | \sigma_s) = P \left( \max_{\sigma_j \in \Sigma \setminus \{\sigma_s\}} |\Delta \mathcal{C}_\gamma^j| \geq |\Delta \mathcal{C}_\gamma^s| \middle| \sigma_s \right), \quad (5)$$

where, for all  $\sigma_j \in \Sigma$ ,

$$\Delta \mathcal{C}_\gamma^j = \psi \circ \mathcal{C}_\gamma(\sigma_j \circ X_1) \cap \mathcal{C}_\gamma(\sigma_s \circ X_2) \quad (6)$$



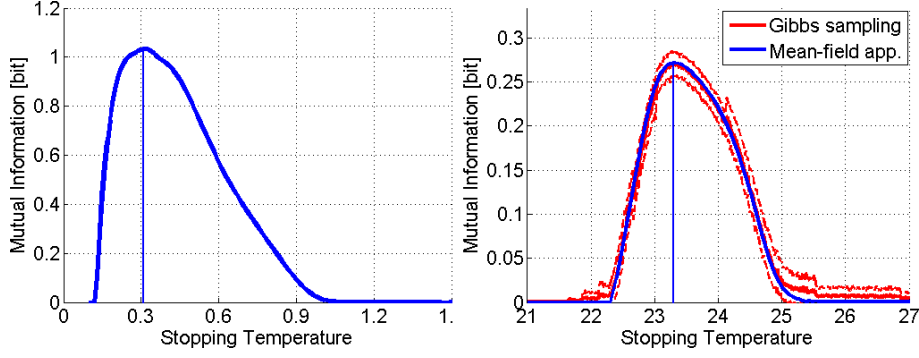


Figure 1. Comparison of the informativeness of pairwise clustering (left) and correlation clustering (right) in terms of AC for gene expression data. The former is approximately four times more informative than the latter. For correlation clustering, the mutual information is estimated by mean-field approximation and Gibbs sampling for comparison.

denotes the intersection between the  $j$ -th approximation set and that of the test set.

The direct evaluation of the error probability can be bounded through the union bound as follows:

$$P(\hat{\sigma} \neq \sigma_s | \sigma_s) \leq \sum_{\sigma_j \in \Sigma \setminus \{\sigma_s\}} P\left(|\Delta \mathcal{C}_\gamma^j| \geq |\Delta \mathcal{C}_\gamma^s| \mid \sigma_s\right), \quad (7)$$

Furthermore, one has that

$$P(\hat{\sigma} \neq \sigma_s) \leq (|\Sigma| - 1) \exp(-n \mathcal{I}_\gamma(\sigma_s, \hat{\sigma})), \quad (8)$$

where  $\mathcal{I}_\gamma(\sigma_j, \hat{\sigma})$  is the mutual information

$$\mathcal{I}_\gamma(\sigma_s, \hat{\sigma}) = \frac{1}{n} \log \left( \frac{|\Sigma| |\Delta \mathcal{C}_\gamma^s|}{|\mathcal{C}_\gamma(X_1)| |\mathcal{C}_\gamma(X_2)|} \right). \quad (9)$$

The optimal  $\gamma$  is found solving

$$\gamma^* = \arg \max_{\gamma \in [0, \infty)} \mathcal{I}_\gamma(\sigma_s, \hat{\sigma}). \quad (10)$$

This procedure provides to the modeler:

- a set of  $\gamma$ -optimal solutions, as well as
- a measure of the informativeness of the selected approximation set for the model  $R$ : the Approximation Capacity (AC)  $\mathcal{I}_\gamma^*(\sigma_s, \hat{\sigma})$ .

This selection criterion enables the comparison of different models  $R$  for the cost of selecting solutions  $c$  given training and test.

### 3. APPLICATIONS AND RESULTS

Recently, ASC has been applied to perform model selection in clustering [12], yielding results consistent with BIC in the analysis of biological data. In clustering,  $\Sigma$  corresponds to the set of permutations of cluster labels. It is worth noting that in the case of clustering the cardinality of the hypothesis class grows exponentially with the sample size. This is because solutions are defined as label assignments in this application.

Experimental results in the context of gene expression analysis show that pairwise clustering [13] yields superior

amounts of reliable information in comparison to correlation clustering [14]. Relational clustering problems are often defined with respect to an attributed graph  $(\mathcal{V}, \mathcal{E})$  with vertex set  $\mathcal{V}$  and edge set  $\mathcal{E}$ . The vertices have to be clustered into groups  $\mathcal{G}_u := \{i : c(i) = u\}, 1 \leq u \leq K$  where  $c$  is the cluster solution which assigns label  $u$  to the  $i$ -th sample. The set of edges between elements of group  $\mathcal{G}_u$  and  $\mathcal{G}_v$  is denoted by  $\mathcal{E}_{uv} := \{(i, j) : c(i) = u \wedge c(j) = v\}$ .

In both cases, the datasets consisted of matrices of pairwise similarities  $X$ . The pairwise clustering cost model is defined as

$$R_{\text{pc}}(c, X) = -\frac{1}{2} \sum_{k=1}^K |\mathcal{G}_k| \sum_{(i,j) \in \mathcal{E}_{kk}} \frac{X_{ij}}{|\mathcal{E}_{kk}|}, \quad (11)$$

where  $X_{ij}$  denotes the similarity between object  $i$  and  $j$ . The correlation clustering model is

$$R^{\text{cc}}(c, X) = \frac{1}{2} \sum_{1 \leq u \leq K} \sum_{(i,j) \in \mathcal{E}_{uu}} (|X_{ij}| - X_{ij}) + \frac{1}{2} \sum_{1 \leq u \leq K} \sum_{1 \leq v < u} \sum_{(i,j) \in \mathcal{E}_{uv}} (|X_{ij}| + X_{ij}).$$

Figure 1 shows the application to gene expression data with temporal structure (expression level time points for 12 consecutive months) [15]. The feature vector is splitted into two and the similarity matrices are constructed by taking the Pearson correlation coefficients for each pair of genes (295 differentially expressed genes). This dataset has been selected because it is one of the many cases in which the choice of a cost is challenging. The figure compares the AC of the two models, showing the advantage of pairwise clustering over correlation clustering. The result means that under identical noise effects, pairwise clustering discovers a more predictive structure than correlation clustering. ASC validates pairwise clustering ( $\max_\beta \mathcal{I}_\beta = 1.03$ , where  $\beta$  is the inverse computational temperature) as approximately 3.5 times more informative than correlation clustering ( $\max_\beta \mathcal{I}_\beta = 0.272$ ). At the optimal resolution (temperature), 7 clusters are discovered by pairwise

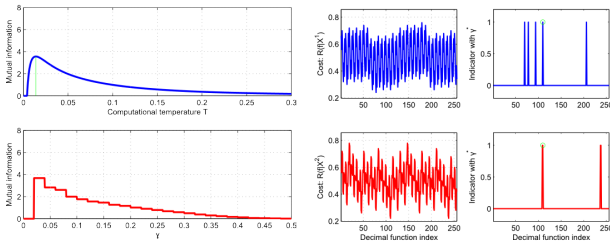


Figure 2. Calculation of mutual information and approximation sets for the Boolean case. On the left, the mutual information is calculated exactly and with the Boltzmann approximation (left top and bottom, respectively). The green line identifies the optimal computational temperature (no normalization). On the right, the model is evaluated for the two split datasets over the hypothesis class (decimal indexing of the Boolean outputs). The green dot indicates the membership of the data generator.

clustering (in contrast to the 2 clusters identified by correlation clustering). The number of clusters in pairwise clustering is also consistent with that obtained with BIC (with number of parameters calculated as the ratio between the trace and the largest eigenvalue of the similarity matrix).

To learn logical propositions we define the hypothesis class of Boolean functions of  $d$  literals. We consider both the supervised and the unsupervised case. In contrast to clustering,  $\Sigma$  is given by the set of distinguishable bit-wise flips of the data (in input for the unsupervised case, and in both input and output in the supervised case). The set of transformations is therefore given by a set of local  $\neg$  (NOT) operators applicable to the available sample components. Hence, in the unsupervised case the cardinality of the set of perturbations is smaller or equal to that of the hypothesis class:

$$|\Sigma| \leq |\mathcal{C}(X)| = 2^{2^d}. \quad (12)$$

The goal is the identification of predictive formulas which generalize the available binary observations. Figure 2 compares the exact solution and Boltzmann approximation with a dataset generated by the 110-th Boolean function with  $d = 3$  subject to uniform sampling of the input. The bit flipping probability is  $1/8$  both for input and for output.

#### 4. ACKNOWLEDGMENTS

We thank Marcus Hutter, Cheng Soon Ong, Ludwig Busse, Brian McWilliams and David Balduzzi for insightful discussions and helpful comments. The authors are financed with grants from the Swiss SystemsX.ch initiative (projects YeastX and LiverX), evaluated by the Swiss National Science Foundation.

#### 5. REFERENCES

[1] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.

[2] A.N. Kolmogorov, “Complexity of algorithms and objective definition of randomness,” in *Talk at Moscow Math. Soc. Meet. (transl. from Russian by L. A. Levin)*, Moscow, Apr. 16 1974.

[3] G.E. Schwarz, “Estimating the dimension of a model,” *Ann. of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[4] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[5] C.S. Wallace and D.M. Boulton, “An information measure for classification,” *Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.

[6] R. Solomonoff, “A formal theory of inductive inference, part i,” *Information and Control*, vol. 7, no. 1, pp. 1–22, 1964.

[7] R. Solomonoff, “A formal theory of inductive inference, part ii,” *Information and Control*, vol. 7, no. 2, pp. 224–254, 1964.

[8] L. Valiant, “A theory of the learnable,” *Comm. of the ACM*, vol. 27, pp. 1134–1142, 1984.

[9] Yevgeny Seldin and Naftali Tishby, “Pac-bayesian analysis of co-clustering and beyond,” *J. Mach. Learn. Res.*, vol. 11, pp. 3595–3646, 2010.

[10] J. M. Buhmann, “Information theoretic model validation for clustering,” in *Proc. of IEEE Int. Symp. on Information Theory 2010*, 2010, pp. 1398–1402.

[11] D. Pál S. Ben-David, U. von Luxburg, “A sober look at clustering stability,” in *Springer Verlag LNAI 4005 Proc. of COLT*, 2006, pp. 5–19.

[12] M. Haghiri Chehreghani, A.G. Busetto, and J.M. Buhmann, “Information theoretic model validation for spectral clustering,” in *J. of Mach. Learn. Res. Proc. of AISTATS 2012*, 2012, pp. 495–503.

[13] T. Hofmann and J.M. Buhmann, “Pairwise data clustering by deterministic annealing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19(1), pp. 1–14, 1997.

[14] S. Chawla N. Bansal, A. Blum, “Correlation clustering,” *Machine Learning*, vol. 56, pp. 89–113, 2004.

[15] F. Mignone H. Boussetta A. Viarengo F. Dondero M. Banni, A. Negri, “Gene expression rhythms in the mussel *Mytilus galloprovincialis* (lam.) across an annual cycle,” *PLoS ONE*, vol. 6(5), pp. e18904, 2011.

# ASYMPTOTIC STATISTICAL ANALYSIS OF STATIONARY ERGODIC TIME SERIES

Daniil Ryabko

INRIA Lille,  
40, avenue Halley  
Parc Scientifique de la Haute Borne  
59650 Villeneuve d'Ascq, France  
daniil@ryabko.net

## ABSTRACT

It is shown how to construct asymptotically consistent efficient algorithms for various statistical problems concerning stationary ergodic time series. The considered problems include clustering, hypothesis testing, change-point estimation and others. The presented approach is based on empirical estimates of the distributional distance. Some open problems are also discussed.

## 1. INTRODUCTION

Statistical problems involving time-series data arise in a variety of modern applications, including biology, finance, network analysis, etc. These applications often dramatically violate traditional statistical assumptions imposed on time series. This applies not only to parametric models, but even to assumptions that are often considered non-parametric, for example that the data points are independent or that the time series have limited memory, or that the processes mix sufficiently fast and so on.

Here I summarize some recent work on statistical analysis of time series where the only assumption on the time series is that they are stationary ergodic. No independence or mixing-type assumptions are involved.

The considered problems are hypothesis testing, clustering, the two- and three-sample problems, and change point estimation. The main results establish asymptotically consistent algorithms for the considered problems. The consistency results follow from the simple fact that the so-called distributional distance [1] can be estimated based on sampling; this contrasts previous results that show that the  $\bar{d}$  distance can not (in general) be estimated for stationary ergodic processes [2]. For more details on these results see [3, 4, 5, 6, 7].

## 2. PRELIMINARIES

Let  $A$  be an alphabet, and denote  $A^*$  the set of tuples  $\cup_{i=1}^{\infty} A^i$ . In this work we consider the case  $A = \mathbb{R}$ ; extensions to the multidimensional case, as well as to more general spaces, are straightforward. Distributions, or (stochastic) processes, are measures on the space  $(A^{\infty}, \mathcal{F}_{A^{\infty}})$ , where  $\mathcal{F}_{A^{\infty}}$  is the Borel sigma-algebra of  $A^{\infty}$ . When talking about joint distributions of  $N$  samples, we mean distributions on the space  $((A^N)^{\infty}, \mathcal{F}_{(A^N)^{\infty}})$ .

For each  $k, l \in \mathbb{N}$ , let  $B^{k,l}$  be the partition of the set  $A^k$  into  $k$ -dimensional cubes with volume  $h_l^k = (1/l)^k$  (the cubes start at 0). Moreover, define  $B^k = \cup_{l \in \mathbb{N}} B^{k,l}$  and  $\mathcal{B} = \cup_{k=1}^{\infty} B^k$ . The set  $\{B \times A^{\infty} : B \in B^{k,l}, k, l \in \mathbb{N}\}$  generates the Borel  $\sigma$ -algebra on  $\mathbb{R}^{\infty} = A^{\infty}$ . For a set  $B \in \mathcal{B}$  let  $|B|$  be the index  $k$  of the set  $B^k$  that  $B$  comes from:  $|B| = k : B \in B^k$ .

We use the abbreviation  $X_{1..k}$  for  $X_1, \dots, X_k$ . For a sequence  $\mathbf{x} \in A^{\infty}$  and a set  $B \in \mathcal{B}$  denote  $\nu(\mathbf{x}, B)$  the frequency with which the sequence  $\mathbf{x}$  falls in the set  $B$ .

$$\nu(\mathbf{x}, B) := \begin{cases} \frac{1}{n-|B|+1} \sum_{i=1}^{n-|B|+1} I_{\{(x_i, \dots, x_{i+|B|-1}) \in B\}} & \text{if } n \geq |B|, \\ 0 & \text{otherwise.} \end{cases}$$

A process  $\rho$  is *stationary* if

$$\rho(X_{1..|B|} = B) = \rho(X_{t..t+|B|-1} = B)$$

for any  $B \in A^*$  and  $t \in \mathbb{N}$ . We further abbreviate  $\rho(B) := \rho(X_{1..|B|} = B)$ . A stationary process  $\rho$  is called (*stationary*) *ergodic* if the frequency of occurrence of each word  $B$  in a sequence  $X_1, X_2, \dots$  generated by  $\rho$  tends to its a priori (or limiting) probability a.s.:

$$\rho\left(\lim_{n \rightarrow \infty} \nu(X_{1..n}, B) = \rho(B)\right) = 1.$$

Denote  $\mathcal{E}$  the set of all stationary ergodic processes.

**Definition 1** (distributional distance). *The distributional distance is defined for a pair of processes  $\rho_1, \rho_2$  as follows (e.g. [1])*

$$d(\rho_1, \rho_2) = \sum_{m,l=1}^{\infty} w_m w_l \sum_{B \in B^{m,l}} |\rho_1(B) - \rho_2(B)|,$$

where  $w_j = 1/j^2$ .

(The weights in the definition are fixed for the sake of concreteness only; we could take any other summable sequence of positive weights instead.) In words, we are taking a sum over a series of partitions into cubes of decreasing volume (indexed by  $l$ ) of all sets  $A^k$ ,  $k \in \mathbb{N}$ , and count the differences in probabilities of all cubes in all these partitions. These differences in probabilities are

weighted: smaller weights are given to larger  $k$  and finer partitions. It is easy to see that  $d$  is a metric. We refer to [1] for more information on this metric and its properties.

The methods below are based on *empirical estimates of the distance  $d$* :

$$\hat{d}(X_{1..n_1}^1, X_{1..n_2}^2) = \sum_{m,l=1}^{\infty} w_m w_l \sum_{B \in B^{m,l}} |\nu(X_{1..n_1}^1, B) - \nu(X_{1..n_2}^2, B)|, \quad (1)$$

where  $n_1, n_2 \in \mathbb{N}$ ,  $\rho \in \mathcal{S}$ ,  $X_{1..n_i}^i \in A^{n_i}$ . Although the expression (1) involves taking three infinite sums, it will be shown below that it can be easily calculated (see Section 4).

### 3. ASYMPTOTIC CONSISTENCY RESULTS

The consistency results are based on the following statement, which is quite easy to derive from the definition of ergodicity (or from Birkhoff's ergodic theorem).

**Lemma 1** ( $\hat{d}$  is consistent). *Let  $\rho_1, \rho_2 \in \mathcal{E}$  and let two samples  $\mathbf{x}_1 = X_{1..n_1}^1$  and  $\mathbf{x}_2 = X_{1..n_2}^2$  be generated by a distribution  $\rho$  such that the marginal distribution of  $X_{1..n_i}^i$   $\rho_i$  is stationary ergodic for  $i = 1, 2$ . Then*

$$\lim_{n_1, n_2 \rightarrow \infty} \hat{d}(X_{1..n_1}^1, X_{1..n_2}^2) = d(\rho_1, \rho_2) \text{ } \rho\text{-a.s.}$$

#### 3.1. The three-sample problem

The first problem we consider is the three-sample problem, also known as process classification. Let there be given three samples  $X = (X_1, \dots, X_k)$ ,  $Y = (Y_1, \dots, Y_m)$  and  $Z = (Z_1, \dots, Z_n)$ . Each sample is generated by a stationary ergodic process  $\rho_X$ ,  $\rho_Y$  and  $\rho_Z$  respectively. Moreover, it is known that either  $\rho_Z = \rho_X$  or  $\rho_Z = \rho_Y$ , but  $\rho_X \neq \rho_Y$ . We wish to construct a test that, based on the finite samples  $X, Y$  and  $Z$  will tell whether  $\rho_Z = \rho_X$  or  $\rho_Z = \rho_Y$ .

The proposed test chooses the sample  $X$  or  $Y$  according to whichever is closer to  $Z$  in  $\hat{d}$ . That is, we define the test  $G(X, Y, Z)$  as follows. If  $\hat{d}(X, Z) \leq \hat{d}(Y, Z)$  then the test says that the sample  $Z$  is generated by the same process as the sample  $X$ , otherwise it says that the sample  $Z$  is generated by the same process as the sample  $Y$ .

**Theorem 1.** *The described test makes only a finite number of errors with probability 1, when  $|X|, |Y|$  and  $|Z|$  go to infinity.*

The statement is easy to derive from Lemma 1. Note that  $X, Y, Z$  are not required to be independent. All we need is that the distributions are stationary ergodic (more formally, the distribution generating the three sequences is arbitrary except for the fact that the marginals are stationary ergodic).

#### 3.2. Time-series clustering

A more general but closely related problem is time-series clustering. We are given  $N$  samples  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , where

each sample  $\mathbf{x}_i$  is a string of length  $n_i$  of symbols from  $A$ :  $\mathbf{x}_i = X_{1..n_i}^i$ . Each sample is generated by one out of  $k$  different *unknown* stationary ergodic distributions  $\rho_1, \dots, \rho_k \in \mathcal{E}$ . Thus, there is a partitioning  $I = \{I_1, \dots, I_k\}$  of the set  $\{1..N\}$  into  $k$  disjoint subsets  $I_j, j = 1..k$

$$\{1..N\} = \cup_{j=1}^k I_j,$$

such that  $\mathbf{x}_j, 1 \leq j \leq N$  is generated by  $\rho_j$  if and only if  $j \in I_j$ . The partitioning  $I$  is called the *target clustering* and the sets  $I_i, 1 \leq i \leq k$ , are called the *target clusters*. Given samples  $\mathbf{x}_1, \dots, \mathbf{x}_N$  and a target clustering  $I$ , let  $I(\mathbf{x})$  denote the cluster that contains  $\mathbf{x}$ .

It is required to partition the index set  $\{1..N\}$  in such a way that as the length of each sequence grows the partitioning coincides with the target clustering from some time on with probability 1. Such an algorithm is called asymptotically consistent. In other words, when the sequences are long enough, we have to group together those and only those sequences that were generated by the same distributions.

This can be done as follows. The point  $\mathbf{x}_1$  is assigned to the first cluster. Next, find the point that is farthest away from  $\mathbf{x}_1$  in the empirical distributional distance  $\hat{d}$ , and assign this point to the second cluster. For each  $j = 3..k$ , find a point that maximizes the minimal distance to those points already assigned to clusters, and assign it to the cluster  $j$ . Thus we have one point in each of the  $k$  clusters. Next simply assign each of the remaining points to the cluster that contains the closest points from those  $k$  already assigned. One can notice that the described algorithm just one iteration of the  $k$ -means algorithm, with so-called farthest-point initialization and using the distance  $\hat{d}$ .

**Theorem 2.** *The described algorithm is strongly asymptotically consistent provided  $\rho_i$  is stationary ergodic for each  $i = 1..k$ .*

#### 3.3. Change-point estimation

Next we consider the change-point problem. The sample  $Z = (Z_1, \dots, Z_n)$  consists of two concatenated parts  $X = (X_1, \dots, X_k)$  and  $Y = (Y_1, \dots, Y_m)$ , where  $m = n - k$ , so that  $Z_i = X_i$  for  $1 \leq i \leq k$  and  $Z_{k+j} = Y_j$  for  $1 \leq j \leq m$ . The samples  $X$  and  $Y$  are generated independently by two different stationary ergodic processes with alphabet  $A = \mathbb{R}$ . The distributions of the processes are unknown. The value  $k$  is called the *change point*. It is assumed that  $k$  is linear in  $n$ ; more precisely,  $\alpha n < k < \beta n$  for some  $0 < \alpha \leq \beta < 1$  from some  $n$  on.

It is required to estimate the change point  $k$  based on the sample  $Z$ .

Note that we do not assume that the single-dimensional marginals before and after the change point are different, as is done almost exclusively in the literature on this problem. We are in the most general situation where the time-series distributions are different, i.e. the change may be only in the long-range dependence.

For each  $t, 1 \leq t \leq n$ , denote  $U^t$  the sample  $(Z_1, \dots, Z_t)$  consisting of the first  $t$  elements of the sample  $Z$ , and denote  $V^t$  the remainder  $(Z_{t+1}, \dots, Z_n)$ .

Define the change point estimate  $\hat{k} : A^* \rightarrow \mathbb{N}$  as follows:

$$\hat{k}(X_1, \dots, X_n) := \operatorname{argmax}_{t \in [\alpha n, n - \beta n]} \hat{d}(U^t, V^t).$$

The following theorem establishes asymptotic consistency of this estimator.

**Theorem 3.** *For the estimate  $\hat{k}$  of the change point  $k$  we have*

$$\frac{1}{n} |\hat{k} - k| \rightarrow 0 \text{ a.s.}$$

where  $n$  is the size of the sample, and when  $k, n - k \rightarrow \infty$  in such a way that  $\alpha < \frac{k}{n} < \beta$  for some  $\alpha, \beta \in (0, 1)$  from some  $n$  on.

This result can be extended [7] to multiple change points and unknown  $\alpha$  and  $\beta$ , although the algorithm becomes much more sophisticated.

### 3.4. Impossibility results: the two-sample problem and its implications

For the problems considered above we have relatively simple algorithms that are asymptotically consistent under most general assumptions. What is more, the proofs of consistency (although mostly omitted here) are quite simple as well. From this one can get the impression that asymptotic consistency results are very easy to obtain and probably they hold for all other interesting problems as well.

This is not the case. The first example is another classical statistical problem: homogeneity testing, also known as the two-sample problem. We are given two samples  $X = (X_1, \dots, X_n)$  and  $Y = (Y_1, \dots, Y_n)$  generated by two stationary ergodic distributions  $\rho_X$  and  $\rho_Y$ . We want to tell whether they were generated by the same or by different distributions, that is, whether  $\rho_X = \rho_Y$ . We are willing to settle for a rather weak asymptotic result. Say a two-sample test  $L(X, Y)$ , that takes two samples and outputs 0 or 1, is asymptotically consistent if  $\mathbf{E}L \rightarrow 1$  as  $n \rightarrow \infty$  if  $\rho_X = \rho_Y$  and  $\mathbf{E}L \rightarrow 0$  otherwise. Moreover, we can further assume that the samples are binary-valued and there is no dependence between  $X$  and  $Y$ . This does not help:

**Theorem 4.** *There is no asymptotically consistent two-sample test.*

This result holds even if we additionally require  $\rho_X$  and  $\rho_Y$  to be  $B$ -processes [5], contrasting earlier results of Ornstein and Weiss for this class of processes [2]. The proof (omitted here) relies on a counterexample which is a limit of hidden Markov processes with a countably infinite state space, using a method similar to that of [8].

As a consequence of this negative result, we can also derive impossibility results for some generalizations of the problems considered above.

**Corollary 1.** *Under the assumptions of theorems 2 and 3 respectively, there is no asymptotically consistent clustering algorithm when the number of clusters is unknown, and there is no asymptotically consistent change-point detection algorithm.*

### 3.5. Hypothesis testing

Some of the problems considered above, as well as many other interesting problems, can be formulated in the following way. Consider two sets  $H_0$  and  $H_1$  which are subsets of the set of all stationary ergodic processes, and let there be given a sample  $X_1, \dots, X_n$  generated by a stationary ergodic process distribution  $\rho$ . We want to tell whether  $\rho \in H_0$  or  $\rho \in H_1$ . The problem arises to characterize those pairs  $(H_0, H_1)$  for which this is possible in some asymptotic sense, that is, whether asymptotically consistent tests exist. It turns out that the distributional distance can be used to answer this question to a considerable extent.

To define the notion of consistency we use for this problem, recall that Type I error is said to occur if the test says “1” while the sample was generated by the distribution from  $H_0$ . Type II error occurs if the test says “0” while  $H_1$  is true. In many practical situations, these errors may have very different meaning: for example, this is the case when  $H_0$  is interpreted as that a patient has a certain ailment, and  $H_1$  that he does not. In such cases, one may wish to treat the errors asymmetrically. Also  $H_0$  can often be much simpler than the alternative  $H_1$ , for example,  $H_0$  can be a simple parametric family, or it may consist of just one process distribution, while  $H_1$  can be the complement of  $H_0$  to the set of all stationary ergodic processes.

Call a test *consistent* if, for any pre-specified level  $\alpha \in (0, 1)$ , any sample size  $n$  and any distribution in  $H_0$  the probability of Type I error (the test says  $H_1$ ) is not greater than  $\alpha$ , while for every distribution in  $H_1$  and every  $\alpha$  the Type II error is made only a finite number of times with probability 1, as the sample size goes to infinity.

Recall that a stationary process can be represented as a mixture of stationary ergodic processes, that is, as a measure on the set  $\mathcal{E}$  (see, e.g., [1]). The set  $\mathcal{E}$  is not closed with respect to the distributional distance, but the set  $\mathcal{S}$  of all stationary process distributions is. The following theorem utilizes these facts. Its proof relies in addition on some other nice properties of the metric space  $(\mathcal{S}, d)$ ; see [6] for the proof and [1] for the properties of  $(\mathcal{S}, d)$ .

**Theorem 5.** *There exists a consistent test for  $H_0$  against  $H_1$  if  $H_0$  has probability 1 with respect to ergodic decomposition of every distribution from the closure of  $H_0$ , where the closure is with respect to the distributional distance  $d$ . Conversely, if there is a consistent test  $H_0$  against  $H_1$  then  $H_1$  has probability 0 with respect to ergodic decomposition of every distribution from the closure of  $H_0$ .*

The necessary and sufficient conditions coincide if  $H_1$  is the complement of  $H_0$  to the set  $\mathcal{E}$  of all stationary ergodic process distributions:

**Corollary 2.** *There exists a consistent test for  $H_0$  against  $H_1 := \mathcal{E} \setminus H_0$  if and only if  $H_1$  has probability 0 with respect to ergodic decomposition of every distribution from the closure of  $H_0$ .*

#### 4. COMPUTATIONAL COMPLEXITY

While the definition of empirical distributional distance  $\hat{d}$  involves taking infinite sums, it can be calculated not only in finite time but efficiently. To see this, first observe that in  $\hat{d}$  all summands corresponding to  $m > n$  (where  $n$  is the min length of  $x_1, x_2$ ) are 0. In the sum over  $l$  (cube size) all the summands are the same from the point where each cube has at most one point in it. This already makes computations finite. Moreover, even though the number of cubes in  $B^{m,l}$  is exponential in  $m$  and  $l$ , at most  $2n$  cubes are non-empty and these are easy to track (across different values of cube size  $l$ ) with a tree structure. Thus,  $\hat{d}$  can be calculated as is (in a naive way) in time  $O(n^2 s \log n)$  where  $s$  is the minimal non-zero distance between points. This can be further reduced: the summands for  $m > \log n$  and for  $l$  such that each cube less than  $\log n$  points have no chance to have consistent estimates and only contribute (a negligible part) to the error. Thus, it is only practical to truncate the sums at  $\log n$ ; since all the theoretical results presented here are asymptotic in  $n$ , it is easy to check that they still hold with this modification of  $\hat{d}$ . The computational complexity of  $\hat{d}$  becomes  $O(n \text{ polylog } n)$ . For more information on implementation of the resulting algorithms see [9]. The latter work also provides some empirical evaluations of the clustering algorithm described here, as well as theoretical results for the online version of this problem.

#### 5. OUTLOOK

Here we mention some interesting open problems for future research. First, the characterisation of those hypotheses for which consistent tests exist is so far incomplete: the necessary and sufficient conditions coincide only in the case when  $H_1$  is the complement of  $H_0$  (cf. Theorem 5 and the corollary). Furthermore, one can consider other notions of consistency of tests, both weaker and stronger ones, such as requiring both probabilities of error to converge to 0, or requiring both errors to be bounded uniformly. An interesting statistical problem that we did not consider here is independence testing. Given two samples it is required to test whether they were generated independently or not. Given the negative result of Theorem 4, one could think that this problem is also impossible to solve. However, Theorem 5 implies that it is, in fact, possible. Finding an actual test (possibly using  $\hat{d}$ ) is an interesting open problem.

#### 6. ACKNOWLEDGMENTS

This work was supported by Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and FEDER through the ‘‘Contrat de Projets Etat Region (CPER) 2007-2013,’’ the ANR projects Explora (ANR-08-COSI-004) and Lampada (ANR-09-EMER-007), by the European Community’s Seventh Framework Program (FP7/2007-2013) under grant agreement 270327 (project CompLACS) and Pascal-2.

#### 7. REFERENCES

- [1] R. Gray, *Probability, Random Processes, and Ergodic Properties*, Springer Verlag, 1988.
- [2] D.S. Ornstein and B. Weiss, ‘‘How sampling reveals a process,’’ *Annals of Probability*, vol. 18, no. 3, pp. 905–930, 1990.
- [3] D. Ryabko, ‘‘Clustering processes,’’ in *Proc. the 27th International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, 2010, pp. 919–926.
- [4] D. Ryabko and B. Ryabko, ‘‘Nonparametric statistical inference for ergodic processes,’’ *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1430–1435, 2010.
- [5] D. Ryabko, ‘‘Discrimination between B-processes is impossible,’’ *Journal of Theoretical Probability*, vol. 23, no. 2, pp. 565–575, 2010.
- [6] D. Ryabko, ‘‘Testing composite hypotheses about discrete ergodic processes,’’ *Test*, vol. 21, no. 2, pp. 317–329, 2012.
- [7] A. Khaleghi and D. Ryabko, ‘‘Multiple change-point estimation in stationary ergodic time-series,’’ Tech. Rep. arXiv:1203.1515v4, arxiv, 2012.
- [8] B. Ryabko, ‘‘Prediction of random sequences and universal coding,’’ *Problems of Information Transmission*, vol. 24, pp. 87–96, 1988.
- [9] A. Khaleghi, D. Ryabko, J. Mary, and P. Preux, ‘‘Online clustering of processes,’’ in *AISTATS*, 2012, JMLR W&CP 22, pp. 601–609.

# BEYOND SHANNON – EXAMPLES FROM GEOMETRY, INFORMATION THEORY AND STATISTICS

Flemming Topsøe

University of Copenhagen  
 Department of Mathematical Sciences  
 Universitetsparken 5, dk-2100 Copenhagen, Denmark, topsoe@math.ku.dk

## ABSTRACT

In previous contributions to WITMSE, [1] and [2], an abstract theory of cognition, inspired by information theory but going beyond classical Shannon theory in certain respects was outlined. See also [3]. Here, we continue the work by presenting three concrete problems: Sylvester’s problem from geometric location theory, a problem of universal coding from information theory and the problem of isotone regression from statistics. At first, we focus on non-technical, philosophically oriented considerations. A more complete analysis of isotone regression follows and finally we point out a surprising connection between this problem and the one from universal coding.

## 1. THREE PROBLEMS

First geometry: In 1857 Sylvester wrote “It is required to find the least circle which shall contain a given system of points in the plane.” In fact, this is the full text of [4]! Thus, if  $X$  denotes the set of points in the plane,  $\|\cdot - \cdot\|$  Euclidean distance and  $\mathcal{P} \subseteq X$  a given system – here assumed finite – of points in  $X$ , we seek a point  $y = y^*$  in  $X$  which minimizes the quantity

$$\max_{x \in \mathcal{P}} \|x - y\|. \quad (1)$$

For the two remaining problems,  $\Omega = (\Omega, \leq)$  denotes a finite partially ordered set provided with a *weight function*  $W$ . Little is lost if you take  $W$  to be the uniform distribution (and this will be assumed if no special mention of  $W$  is made). A real-valued function  $f$  on  $\Omega$  is *isotone* if, for  $a, b \in \Omega$ , the implication  $a \leq b \Rightarrow f(a) \leq f(b)$  holds. And  $f$  is *antitone* if  $-f$  is isotone.

The problem from information theory which we shall deal with concerns the *model*  $\mathcal{A}$  of all antitone probability distributions over  $\Omega$ . Requested is the distribution  $y = y^*$  which best represents  $\mathcal{A}$  in the sense that

$$\sup_{x \in \mathcal{A}} D(x||y) \quad (2)$$

is minimized. Here  $D$  stands for *Kullback-Leibler divergence*, i.e.  $D(x||y) = \sum_{a \in \Omega} x(a) \ln \frac{x(a)}{y(a)}$ . This is a problem of *universal prediction*.

The corresponding problem of *universal coding* is to find a suitable *code length function* (in the sequel simply a *code*),  $\kappa^*$ , which can be taken as the base for actual coding of observations from a source emitting independent outputs from  $\Omega$ , generated by a distribution known only to lie in  $\mathcal{A}$ . Appealing to standard information theoretical insight, the sought *universal code* is  $\kappa^*$  given from  $y^*$  by  $\kappa^*(a) = \ln \frac{1}{y^*(a)}$  for  $a \in \Omega$  (the good sense of this also involves an idealization and a replacement of logarithms to the base 2 with natural logarithms). Our codes satisfy *Kraft’s equality*:  $\sum_{a \in \Omega} \exp(-\kappa(a)) = 1$ .

As our final problem we take *isotone least squares regression* (below just *isotone regression*), an important problem from statistics. Given is a real-valued function  $y_0$  on  $\Omega$ , referred to as a *valuation*. Sought is the isotone valuation  $y = y^*$  which is closest in mean-squared norm to the given valuation  $y_0$ . Thus, we should minimize

$$\|y_0 - y\|^2 = \sum_{a \in \Omega} W(a) |y_0(a) - y(a)|^2 \quad (3)$$

subject to a requirement on  $y$  of isotonicity. Just as with the two previous problems, existence and uniqueness of the sought object is pretty evident. We refer to it as the *isotone regression of  $y_0$*  (or just the *isotone regression*).

## 2. A COMMON FRAMEWORK

There exists a common framework which allows an efficient treatment of problems as those presented and of many others – e.g. from information theory, one could point to problems of maximum entropy determination, information projections and capacity determination. The reader is referred to [1] and [2] (or to a more comprehensive study, not yet in final form). Rather than spending time here on technicalities, we shall emphasize some features of the underlying theory as seen in the light of the three problems above.

The problems presented are all *optimization problems*. The first two are quite similar, technically. Euclidean distance stands out for the first, Kullback-Leibler divergence for the second. One should, however, note that optimization as in (1) and (2), does not uniquely tell us which are the basic quantities as any strictly increasing function of the appearing quantities could also be used. As we

shall argue below – and not all that surprising – squared Euclidean distance is adequate for the first problem and Kullback-Leibler divergence itself for the second.

A guiding principle for the choice of appropriate basic quantities is that – as recognized since long in optimization theory and convex analysis – one benefits from treating along with a given problem, also a *dual* problem. For this to work out conveniently, one needs certain strict relationships to hold which essentially involve conditions of linearity or affinity. Theoretically, introductory considerations can be carried out without imposing such strict conditions, cf. [1] and [2]. However, when it comes to actually treating concrete problems of interest, you need to be more specific.

In order to motivate necessary restrictions for a successful model building, we claim that the “two-ness” of duality considerations is best expressed by choosing a game-theoretical setting involving certain asymmetric *two-person zero-sum games*. For these games, the players have quite different roles. The first player, considered female, is conceived as “*Nature*”. Nature chooses a strategy which reflects “*truth*”, whereas the second player is a much more easily understood being, “*you*” or “*Observer*” – a mere mortal person, male we reckon, seeking the truth but restricted to “*belief*”. Analyzing these thoughts, you find that though tempting to imagine Nature as a rational being reflecting “*absolute truth*”, really, this is naive and what is involved is more sensibly thought of as another side of yourself. The “*zero-sumness*” of the games you are led to consider express an insight consistent with ideas of Jaynes from the mid-fifties, cf. [5], viz. that acting in a way which would contradict the zero-sum character would reflect that “you have known something more” and, therefore, your model building would be incomplete and should be adjusted.

An essential restriction in our model building then is that the games considered should, typically, be in *equilibrium*, i.e. the *minimax* and *maximin* values should coincide. In many cases this is not so at first sight. E.g., for the two first problems, where a minimax-value is sought, we find that the corresponding maximin-value is uninformative, indeed it vanishes identically. This may be remedied if suitable extensions of the allowed strategies for Nature can be devised. For the two problems pointed to, this can be achieved by allowing *randomized strategies* for Nature (and, regarding (1), replacing norm by squared norm). In this way a common game theoretical base for the treatment of these problems can be found. This also applies to the third problem, though it is of a different type. There it pays to consider the given valuation  $y_0$  as a parameter, cf. Section 3.

One has to be realistic as to what can be expected of a common theoretical base. In fact, though problems we are able to deal with typically have unique solutions, e.g. none of the three concrete problems considered allow solutions in closed form. One has to be satisfied with numerical algorithms or turn to special cases where solutions can be written down in closed form or, more realistically,

where finite state algorithms of low complexity leads to the solution. Such algorithms are special. Often Galois theory shows that even rather “small” problems have solutions which cannot be expressed quantitatively using the basic algebraic operations applied to the natural quantitative specifications of the problems.

Thus, an appeal to game theory does not in itself lead to solutions of the problems at hand. But it does help to characterize what is required of a solution. Such results of *identification* are often derived from an application of the *saddle-value inequalities* now associated with Nash’s name. An example of this follows in the next section.

The overall theme of our investigations, that of establishing a useful theoretical base going “beyond Shannon”, has been pursued by several authors in one way or another and appears right now to be gaining momentum, cf. also [6]. Shannon himself was aware of the need to broaden the theory he had initiated, e.g., in 1953 he writes “It is hardly to be expected that a single concept of information would satisfactorily account for the numerous possible applications of this general field”, cf. [7].

### 3. ISOTONE REGRESSION

Let us leave the airy considerations of the foregoing section and turn to a closer study of isotone regression. The key to a game-theoretical formulation is the binary function  $U_{|y_0} = U_{|y_0}(x, y)$  given by

$$U_{|y_0}(x, y) = \|x - y_0\|^2 - \|x - y\|^2. \quad (4)$$

This is interpreted as the *updating gain*, when the *prior*  $y_0$  is updated by Observers choice of the *posterior*  $y$ , assuming that the strategy chosen by Nature is  $x$ . In (4),  $x$  runs over the set  $X$  of all isotone valuations. These are the strategies of Nature. The strategies of Observer may be taken to be the set of all valuations, but it may also be restricted to  $X$ .

If Nature chooses  $x$ , the best response by Observer is also to choose  $x$ . The resulting value of  $U_{|y_0}$  will then be  $\|x - y_0\|^2$  and it follows that the *optimal strategy* for Nature is to choose the sought isotone regression.

Comparing with Section 3 of [2], you realize that all conditions stated there are fulfilled. In particular, the squared norm satisfies the *compensation identity* (13) of [2]. From Theorems 2 and 3 of [2], it follows that Nature and Observer both have unique optimal strategies  $x^*$  and  $y^*$  and that these strategies coincide:  $x^* = y^*$ . A key problem is, therefore, to determine this common *bi-optimal strategy*. A suitable result of identification for this problem will now be derived.

Let  $x^* = y^*$  be a given isotone valuation, from the outset not known to be the sought bi-optimal strategy. Then, by the general theory, this *is* the sought strategy if and only if the non-trivial part of Nash’s inequalities holds:

$$U_{|y_0}(\xi, y^*) \geq \|x^* - y_0\|^2 \text{ for every } \xi \in X. \quad (5)$$

Expressing squared norm via the associated inner product defined by  $\langle f, g \rangle = \sum_{a \in \Omega} W(a)f(a)g(a)$ , and recalling that  $y^* = x^*$ , we transform the requirement to the



condition

$$\langle \xi - x^*, x^* - y_0 \rangle \geq 0 \text{ for every } \xi \in X. \quad (6)$$

For the further analysis, we note that any valuation  $f$  induces a special decomposition of  $\Omega$ , denoted  $\mathcal{S}_f$ . The sets in  $\mathcal{S}_f$  are the *maximal connected sets of  $f$ -constancy*, i.e. the connected subsets of  $\Omega$  on which  $f$  assumes the same value and which are maximal with respect to these properties. Further, we note that in case  $f$  is isotone, the sets in  $\mathcal{S}_f$  are partially ordered in a natural way, viz. by defining  $A < B$  to mean that, firstly,  $A \neq B$  and, secondly, that  $a < b$  for some  $(a, b)$  with  $a \in A$  and  $b \in B$ .

Any valuation  $f$  is specified by the decomposition  $\mathcal{S}_f$  and the associated function values. For the isotone regression only the decomposition  $\mathcal{S}^* = \mathcal{S}_{x^*}$  needs to be specified as the function values can then be identified as *conditional averages*. Indeed, denoting by  $\bar{A}_{|y_0}$  (or simply  $\bar{A}$ ) the conditional average of the prior  $y_0$  over  $A$ , i.e.

$$\bar{A} = \sum_{a \in A} W(a|A)y_0(a) = \frac{1}{W(A)} \sum_{a \in A} W(a)y_0(a), \quad (7)$$

then, for the isotone regression  $x^*$ ,

$$\text{for all } A \in \mathcal{S}^*, x^* = \bar{A} \text{ on } A. \quad (8)$$

In fact, this is easy to prove by a differential argument based on the considerations of valuations obtained from  $x^*$  by varying the value on  $A$  and keeping other values fixed. The argument can be refined, yielding another central property of  $\mathcal{S}^*$ , *boundedness*. This is the property, that for each  $A \in \mathcal{S}^*$  and each *lower set*  $L$  which intersects  $A$  – a lower set being a set such that  $a < b \in L$  implies  $a \in L$  – it holds that

$$\bar{A}_{|y_0} \leq \overline{A \cap L}_{|y_0}. \quad (9)$$

**Theorem 1 (Identification)** *Let  $x$  be a valuation with associated decomposition  $\mathcal{S}$  and associated function-values  $\alpha(A)$ ;  $A \in \mathcal{S}$ . Then a necessary and sufficient condition that  $x = x^*$ , the sought isotone regression of  $y_0$ , is that the following conditions hold: (i) [ordering]:  $\mathcal{S}$  is partially ordered; (ii) [monotonicity]: if  $A, B \in \mathcal{S}$  and  $A < B$ , then  $\alpha(A) < \alpha(B)$ ; (iii) [proper values]:  $\alpha(A) = \bar{A}_{|y_0}$  for each  $A \in \mathcal{S}$  and (iv) [boundedness]: for every  $A \in \mathcal{S}$  and every lower set  $L$  which meets  $A$ , (9) holds.*

**Proof** A proof that the stated conditions are necessary was indicated above. In order to establish sufficiency, assume that the conditions hold. The essential point is to establish the validity of (6). An indication has to suffice: First, write the inner product in (6) as a sum and then split the sum in a sum over each of the classes in  $\mathcal{S}$ . For the essential argument we may assume that  $\mathcal{S} = \{\Omega\}$ . Consider a fixed isotone valuation  $\xi$ . Let  $\alpha_0 < \dots < \alpha_n$  be the values assumed by  $\xi$  and write  $\xi$  in the form

$$\xi = \alpha_n - \sum_{i=1}^n (\alpha_i - \alpha_{i-1}) 1_{\{\xi < \alpha_i\}}. \quad (10)$$

Consider the valuation  $\delta$  defined by

$$\delta(a) = W(a)(\bar{\Omega}_{|y_0} - y_0(a)). \quad (11)$$

Then  $\sum_{a \in \Omega} \delta(a) = 0$  and  $\sum_{a \in L} \delta(a) \leq 0$  for each lower set  $L$ . By (10) it follows that  $\sum_{a \in \Omega} \xi(a)\delta(a) \geq 0$ , which is the required result.  $\square$

A discussion is in order. The reasoning demonstrates that though Nash's inequalities in principle contain the essentials, this may be in a somewhat concealed form and require quite a bit of extra work until a transformation into a manageable form has been obtained. We may also note that though the identification result is easy to use in examples of moderate size – see, e.g. the butterfly set discussed in Figures 1 and 2 – the necessary checking of condition (iv) of Theorem 1 may be forbidding for more elaborate partially ordered sets as the number of lower sets may be of exponential size in the number of parameters necessary to specify the partial order.

Thus one should ask for further results aiming at the actual construction of the isotone regression. Often, this is not feasible but, fortunately, the problem dealt with is one for which satisfactory results exist, cf. [8] and references referred to there, especially [9].

The problem is greatly simplified if we restrict attention to tree-like structures. We shall assume from now on that  $\Omega$  is a *co-tree*, i.e. right sections are well ordered (or, equivalently, the reverse partial ordering is a tree). This is a significant simplification. For one thing, lower sets can then be represented as disjoint unions of left sections, thus the checking involved in the identification theorem is feasible, as only left sections need to be checked when checking the boundedness property.

Without being very specific, the existence of an efficient algorithm for the determination of the isotone regression is indicated below. The ideas are contained in the identification theorem. As it turns out, if you focus on all properties *except* boundedness and aim at construction of the classes in  $\mathcal{S}^*$  “from below”, then an argument (not shown here) will reveal the fact that boundedness is verified automatically. The build-up from below exploits the idea of searching for violation of the monotonicity requirement followed by pooling of adjacent already constructed classes if a violation occurs. This idea is well known from the statistical literature on isotone regression and there referred to as *pooling of adjacent violators* (PAV). The example of a linear ordering as displayed in Figure 3 explains better than many words how the intended algorithm works. And generalizing to an arbitrary co-tree presents no further problems.

#### 4. A SURPRISING CONNECTION

Consider again the problem of universal coding. The assumption, still in force, that  $\Omega$  is a co-tree, implies that the model  $\mathcal{A}$  is a simplex with the uniform distributions over left sections as extremal elements. Denote by  $a^\perp$  the left section determined by  $a$ , by  $N(a)$  the number of elements in  $a^\perp$  and by  $U_a$  the uniform distribution over  $a^\perp$ . Further, let  $a^-$  be the set of immediate predecessors of  $a$ .

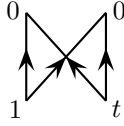


Figure 1. Butterfly with valuation depending on a parameter  $t$ .

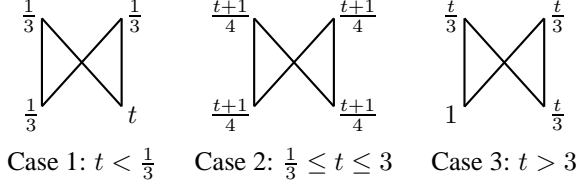


Figure 2. Isotone regression for the butterfly, depending on the value of the parameter  $t$ .

It is easy to check that there exists a distribution  $Q$ , not necessarily isotone, such that  $D(U_a||Q)$  is independent of  $a$ . Indeed,  $Q$  is proportional to  $\mu$  given by

$$\mu(a) = \frac{\prod_{b \in a^-} N(b)^{N(b)}}{N(a)^{N(a)}}, \quad a \in \Omega. \quad (12)$$

**Theorem 2** Let  $y_0$  be the valuation given by

$$y_0(a) = \ln \frac{1}{\mu(a)}; \quad a \in \Omega, \quad (13)$$

and denote by  $y^*$  the isotone regression of  $y_0$ . Then the universal code  $\kappa^*$  is obtained from  $y^*$  by normalization, i.e., for a suitable constant,  $c$ ,  $\kappa^*(a) = y^*(a) + c$  for every  $a \in \Omega$ .

This follows, in a rather roundabout manner, by comparing [10] with results from isotone regression. A more direct proof may well exist.

The special distribution  $Q$  with constant divergence to a set of elements which generate the relevant model may be called a *Sylvester point*. It is easy to see that the universal predictor can be obtained as the information projection of  $Q$  on the model  $\mathcal{A}$ . Analogous features apply to Sylvester's problem, though the existence of a Sylvester point in that setting is only possible in very special cases, e.g. for the illuminating case of a three-element model  $\mathcal{P}$ .

## 5. ACKNOWLEDGMENTS

This goes to Henrik Densing Petersen, cf. [10], to a referee of [10] who encouraged a comparison with [8] and to Peter Harremoës for a discussion of the boundedness property of Theorem 1.

## 6. REFERENCES

[1] F. Topsøe, "Cognition beyond Shannon," in *Proceedings of the third Workshop on Information Theoretic Methods in Science and Engineering, Tampere, 2010*, available from <http://sp.cs.tut.fi/WITMSE10/Proceedings/index.html>.

7								<del>8</del>	8
9						9	9	9	8
6						6	6	6	6
3			<del>5</del>	<del>4</del>	4	4	4	4	4
5		5	5	<del>4</del>	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4
start $y_0$		5	4				9	so-	lu-
		violates	violates				violates	tion	
		-	-				-		
		pool	pool				pool		
		it!	it!				it!		$y^*$

Figure 3. Algorithmic construction of the isotone regression for a 6-element linear order with valuation  $y_0 = (4, 5, 3, 6, 9, 7)$ .

- [2] F. Topsøe, "Cognition and Inference in an Abstract Setting," in *Proceedings of the fourth Workshop on Information Theoretic Methods in Science and Engineering, Helsinki, 2011*, Report C-2011-45, pp. 67–70, University of Helsinki, available from <http://www.helsinki.fi/witmse2011/proceedings.html>.
- [3] F. Topsøe, "Game Theoretical Optimization inspired by Information Theory," *J. Global Optim.*, pp. 553–564, 2009.
- [4] J. J. Sylvester, "A question in the geometry of situation," *Quarterly Journal of Pure and Applied Mathematics*, vol. 1, pp. 79, 1857.
- [5] E. T. Jaynes, *Probability Theory - The Logic of Science*, Cambridge University Press, Cambridge, 2003.
- [6] W. Szpankowski, "Algorithms, Combinatorics, Information, and Beyond," *IEEE Information Theory Society Newsletter*, vol. 62, pp. 5–20, 2012.
- [7] C. Shannon, "The lattice theory of information," *IRE professional Group on Information Theory*, vol. 1, pp. 105–107, 1953.
- [8] P.M. Pardalos and G. Xue, "Algorithms for a Class of Isotonic Regression Problems," *Algorithmica*, vol. 23, no. 3, pp. 211–222, Mar. 1999.
- [9] C. I. C. Lee, "The Min-Max Algorithm and Isotonic Regression," *Ann. Statist.*, vol. 11, pp. 467–477, 1983.
- [10] H. D. Petersen and F. Topsøe, "Computation of universal objects for distributions over co-trees," under publication in *IEEE Trans. Inform. Theory*, 2012.

# CHANGE DETECTION, HYPOTHESIS TESTING, AND DATA COMPRESSION

Kenji Yamanishi<sup>1</sup>, Ei-ichi Sakurai<sup>2</sup>, Hiroki Kanazawa<sup>1</sup>

<sup>1</sup>Graduate School of Information Science and Engineering, The University of Tokyo,  
yamanishi@mist.i.u-tokyo.ac.jp, hiroki\_kanazawa@mist.i.u-tokyo.ac.jp

<sup>2</sup>National Institute of Advanced Industrial Science and Technology, e.sakurai@aist.go.jp

## ABSTRACT

We are concerned with the issue of detecting changes of statistical models when they change over time. We introduce the dynamic model selection (DMS) algorithm for learning model sequences on the basis of the minimum description length (MDL) principle. We first analyze it from the view of hypothesis testing. We evaluate error probabilities for testing the occurrences of change-points and relate them to the model transition estimators and the distance between the models to be distinguished. We then apply the DMS algorithm into data compression via piecewise stationary memoryless sources (PSMS's). We give a method for discretizing the parameter space to obtain an optimal data compression bound. From the both views of hypothesis testing and data compression, we argue how to discretize the parameter space in order to obtain ideal performance. It yields a new view of distinguishability of probabilistic models from the standpoint of change-detection.

## 1. INTRODUCTION

We are concerned with the issue of detecting changes of probabilistic models from a non-stationary data sequence. Dynamic model selection, which we abbreviate as DMS, has been proposed in [14],[13](see also [3]) in order to address this issue. DMS algorithms have been designed on the basis of the minimum description length (MDL) principle ([8]). I.e., they output a model sequence so that the sum of the code-length for a data sequence plus that for a model sequence is minimum. DMS is related to works by van Erven et.al.[2] on switching distributions, those by Shamir and Merhav [10], Willems [11], Willems and Casadei [12] on data compression for piecewise stationary memoryless sources (PSMSs).

In this paper we first analyze DMS from the view of hypothesis testing. We apply DMS to the issue of testing whether a change-point of statistical models exists or not, and evaluate it in terms of Type 1 and 2 error probabilities, which depend on how to estimate model transitions. We investigate them for the three types of methods for estimating model transition probabilities: Shamir and Merhav's method [10], Krichevsky and Trofimov's one [6], and Willem's one [11],[12].

We then apply DMS to data compression. We derive upper bounds on the total code-length for the three meth-

ods for estimating model transitions. We also apply DMS to learning piecewise stationary memoryless sources (PSMSs[9]) and analyze it from the view of data compression. According to [4], we give a method for discretizing the parameter space in order to get an optimal code-length bound. From the both views of hypothesis testing and data compression, we argue how to discretize the parameter space to obtain ideal performance. This yields a new insight into distinguishability([1],[8]) of probabilistic models from the view of change-detection as well as data compression.

## 2. DYNAMIC MODEL SELECTION

Following [14],[13] we introduce a framework for DMS. Let  $\mathcal{X}$  be a domain, which may be either continuous or discrete. Let  $x$  take a value in  $\mathcal{X}$ . Let  $\mathcal{M}$  be a class of models, each of which is specified by a discrete parameter and is properly ordered. For example, we may consider the case where  $M \in \mathcal{M}$  is a dimension of real-valued parameters. We denote  $x_1 \dots x_{t-1}$  as  $x^{t-1}$ . Let  $P(X^n|M)$  be a probability distribution specified by a model  $M$ . For each  $M$ , for each  $t$ , we define a predictive distribution of  $X$  given  $x_a^b$  by  $P(X|x^{t-1} : M) = P(X \cdot x^{t-1}|M)/P(x^{t-1}|M)$ .

We suppose that a model switches to neighboring ones with some probabilities at each time. According to [14], we consider model transition probability distributions:

**Definition 1** Let  $M$  range over  $\{1, \dots, \bar{M}\}$ . Let  $\alpha$  be a 1-dimensional parameter. Assuming that a model transits to neighbouring ones only, we define the *model transition probability distribution* as:

$$P(M_t|M^{t-1} : \alpha) = \begin{cases} 1 - \alpha & \text{if } M_t = M_{t-1}, M_t \neq 1 \text{ or } \bar{M}, \\ 1 - \alpha/2 & \text{if } M_t = M_{t-1}, M_t = 1 \text{ or } \bar{M}, \\ \alpha/2 & \text{if } |M_t - M_{t-1}| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Here are three methods for estimating  $\alpha$ .

**Definition 2** Shamir and Merhav's (SM) estimator  $\hat{\alpha}$  is defined as follows[10]: For  $\epsilon > 0$ ,

$$\hat{\alpha}(M^t) = \frac{\pi(t - t_c + 1)}{Z_\infty - Z_{t-t_c}}, \quad (1)$$

where  $t_c$  is the latest change point before  $t$  and  $\pi(t) = \frac{1}{t^{1+\epsilon}}$ ,  $Z_n = \sum_{j=1}^n \pi(j)$ ,  $Z_\infty = \sum_{j=1}^\infty \pi(j)$ . Krichevsky and Trofimov's (KT) estimator  $\hat{\alpha}$  is defined as follows[6]:

$$\hat{\alpha}(M^t) = (n(M^t) + 1/2)/t, \quad (2)$$

where  $n(M^t)$  is the number of model changes in  $M^t$ . Willem's ( $W$ ) estimator  $\hat{\alpha}$  is defined as follows[11]:

$$\hat{\alpha}(M^t) = 1/(2(t - t_c)), \quad (3)$$

where  $t_c$  is the latest change-point before  $t$ .

KT estimator is calculated using all the past data, while SM and W estimators are calculated using the data starting from the latest change-point.

We denote  $P(M_t|M^{t-1} : \hat{\alpha}(M^{t-1}))$  as  $\hat{P}_t(M_t|M^{t-1})$ . Below we give a criterion for selecting an optimal sequence on the basis of the MDL principle.

**Definition 3** [14] Given  $x^n = x_1 \dots x_n$ , we define the *DMS criterion* for  $M^n = M_1 \dots M_n$  by:

$$\begin{aligned} \ell(x^n : M^n) &= \sum_{t=1}^n (-\log P(x_t|x^{t-1} : M_t)) \\ &+ \sum_{t=1}^n \left( -\log \hat{P}_t(M_t|M_{t-1}) \right). \end{aligned} \quad (4)$$

The first term is the total predictive code-length for  $x^n$  relative to  $M^n$  while the second term is the total predictive code-length for  $k^n$ . Hence the optimal sequence is obtained as the one which minimizes the total code-length. It leads to the DMS algorithm as follows:

**Definition 4** [14] *The DMS algorithm*, denoted as DMS, is an algorithm that takes as input  $x^n$  and outputs  $\hat{M}^n$  s.t.

$$\hat{M}^n = \arg \min_{M^n} \ell(x^n : M^n). \quad (5)$$

An algorithm that computes  $\hat{M}^n$  as in (5) using the dynamic programming has been proposed ([14]).

### 3. HYPOTHESIS TESTING WITH DMS

We simplify the problem of DMS so that there are only two models;  $M_1$  and  $M_2$ . We are then concerned with the issue of testing whether a model has changed or not. Below we assume that the model is either  $M_1$  or  $M_2$ , the initial model is  $M_1$ , and there exists only one change-point in a model sequence. The problem is to detect when the model has changed. We give the following specific form of DMS in order to solve this issue.

**Definition 5** *DMS as a change-point detector* is an algorithm that takes as input  $x^n$  and outputs the least time index  $t_c$  such that

$$\ell(x^n : M_1^n) \geq \ell(x^n : M^n(t_c)), \quad (6)$$

where  $M^n(t_c) \stackrel{\text{def}}{=} \overbrace{M_1 \dots M_1}^{t_c} \overbrace{M_2 \dots M_2}^{n-t_c}$  and  $M_1^n \stackrel{\text{def}}{=} M_1 \dots M_1$ .

We reduce the change-detection problem to the hypothesis testing as follows: Let  $t^*$  be a true change-point. Consider the following two hypotheses:  $H_0$  and  $H_1$ :

$$\begin{aligned} H_0 : & M_1 \quad \text{for } x_1^n = x_1 \dots x_n, \\ H_1 : & \begin{cases} M_1 & \text{for } x_1^{t^*} = x_1 \dots x_{t^*}, \\ M_2 & \text{for } x_{t^*+1}^n = x_{t^*+1} \dots x_n. \end{cases} \end{aligned}$$

Set  $P(x_{t^*+1}^n|x^{t^*} : M_1) \stackrel{\text{def}}{=} \prod_{j=t^*+1}^n P(x_j|x^{j-1} : M_1)$ , and  $P(x_{t^*+1}^n|x^{t^*} : M_2) \stackrel{\text{def}}{=} \prod_{j=t^*+1}^n P(x_j|x^{j-1} : M_2)$ . Then

DMS as a change-point detector works as a hypothesis testing algorithm such that  $H_0$  is accepted if

$$\begin{aligned} & \sum_{t=t^*+1}^n (-\log P(x_t|x^{t-1} : M_1)) \\ & - \sum_{t=t^*+1}^n (-\log P(x_t|x^{t-1} : M_2)) < f(n, t^*), \end{aligned} \quad (7)$$

where

$$f(n, t^*) \stackrel{\text{def}}{=} \ell(M^n(t^*)) - \ell(M_1^n), \quad (8)$$

and

$$\begin{aligned} \ell(M_1^n) &\stackrel{\text{def}}{=} \sum_{t=1}^n (-\log \hat{P}_t(M_1|M_1)), \\ \ell(M^n(t^*)) &\stackrel{\text{def}}{=} \sum_{t=1}^{t^*-1} (-\log \hat{P}_t(M_1|M_1)) + (-\log \hat{P}_{t^*}(M_2|M_1)) \\ &+ \sum_{t=t^*+1}^n (-\log \hat{P}_t(M_2|M_2)). \end{aligned}$$

Otherwise  $H_1$  is accepted.

We define as measures of performance of a change-point detector Type 1 and 2 error probabilities as follows:

**Definition 6** For given the length of data sequence  $n$ , the change-point time  $t^*$ , we define *Type 1 error probability* for DMS as a change-point detector by:

$$\text{Prob} [x_{t^*+1}^n \sim P(X^n|M_1) \text{ and Eq.(7) doesn't hold} ],$$

and *Type 2 error probability* for DMS at delay  $h = n - t^*$  by:

$$\text{Prob} [x_{t^*+1}^n \sim P(X_{t^*+1}^n|x^{t^*} : M_2) \text{ and Eq.(7) holds} ].$$

Type 1 error probability is the probability that the model change has not yet occurred until time  $n$  but the change is incorrectly reported at time  $t^*$ . Type 2 error probability is the probability that the model change has already occurred at time  $t^*$ , but it is overlooked until time  $n$  where  $h = n - t^*$  is *detection delay*.

We make the following assumption for  $M_1$  and  $M_2$ .

**Assumption 7** Suppose that for some  $0 < K < \infty$ , for any  $X$ ,  $|\log P(X|M_i)| \leq K$  for  $i = 1, 2$  and that for some  $0 < V < \infty$  the variance of the random variable  $V_j = \log P(X_j|X^{j-1} : M_2)/P(X_j|X^{j-1} : M_1)$  with respect to  $P(X_j|X^{j-1} : M_2)$  is upper-bounded by  $V$  for any  $j$ .

We give the following theorem on Type 1 and 2 error probabilities for general cases.

**Theorem 8** For DMS as a change-point detector, we have

$$\text{Type 1 error probability} \leq 2^{-f(n, t^*)}. \quad (9)$$

Let us define the Kullback-Leibler divergence (the KL-divergence) between  $P(X^h|x^{t^*} : M_2)$  and  $P(X^h|x^{t^*} : M_1)$  by

$$\begin{aligned} & D_h(M_2||M_1)|_{x^{t^*}} \\ & \stackrel{\text{def}}{=} \sum_{X_{t^*+1}^n} P(X_{t^*+1}^n|x^{t^*} : M_2) \log \frac{P(X_{t^*+1}^n|x^{t^*} : M_2)}{P(X_{t^*+1}^n|x^{t^*} : M_1)}. \end{aligned}$$

Under Assumption 7, if  $D_h(M_2||M_1)|_{x^{t^*}} > f(n, t^*)$  holds, for some  $0 < C < \infty$ , we have

$$\text{Type 2 error probability} \leq 2 \exp(-Ch\beta_h^2), \quad (10)$$

where

$$\beta_h \stackrel{\text{def}}{=} \frac{1}{h} (D_h(M_2||M_1)|_{x^{t^*}} - f(n, t^*)), \quad (11)$$

Theorem 8 shows that Type 1 error probability for DMS is always upper-bounded by the exponential in the negative  $f(n, t^*)$ , which is determined by only the code-lengths for model transition. We also see that Type 2 error probability for DMS decays in order  $O(\exp(-h\beta_h^2))$ , where the exponent factor depends on the code-length for model transition as well as the KL-divergence between  $M_2$  and  $M_1$ . The larger the KL-divergence minus  $f(n, t^*)$  is, the smaller Type 2 error probability is. The larger  $f(n, t^*)$  is, the smaller Type 1 error probability is while the larger Type 2 error probability is. The balance between Type 1 and 2 error probabilities depends on how to estimate model transition probability distributions. We have the following corollaries for the respective model transition estimators.

**Corollary 9** *Let the values of  $f(n, t_*)$  as in (8) for SM estimator, KT estimator and Westimator be  $f^{SM}(n, t^*)$ ,  $f_{KY}(n, t^*)$ , and  $f^W(n, t^*)$ , respectively. Then they are given as follows:*

$$f^{SM}(n, t^*) = \log Z_\infty t^{*(1+\epsilon)} + \log \left\{ \left( \frac{h+1}{h+1+\epsilon} \right) \left( \frac{h+1}{n} \right)^\epsilon \right\},$$

$$f^{KT}(n, t^*) = \log(2(t^* + h) - 1),$$

$$f^W(n, t^*) = \log \frac{(n-1/2)_h h!}{n_h (h-1/2)_h} + \log(2t^* - 1),$$

where  $(n-1/2)_h = (n-1/2)(n-3/2)\cdots(t^*+1/2)$  and  $n_h = n(n-1)\cdots(t^*+1)$ .

We may see that for fixed  $t^*$ , for sufficiently large  $h$  for sufficiently small  $\epsilon > 0$ ,

$$f^{KT}(n, t^*) > f^{SM}(n, t^*) > f^W(n, t^*). \quad (12)$$

This implies that Type 1 error probability becomes small in this order while Type 2 error probability becomes large in this order.

## 4. DATA COMPRESSION WITH DMS

### 4.1. Data Compression

When we apply DMS of Definition 4 into data compression, we have the following theorem on its total code-length:

**Theorem 10** *For any  $x^n$ , the total code-length for DMS, which we denote as  $\ell(x^n)$ , is upper-bounded as follows:*

$$\ell(x^n) \leq \min_m \min_{t_0, \dots, t_m} \min_{M(0), \dots, M(m)} \left\{ \log |\mathcal{M}| + F(n, m) + \sum_{j=0}^m \sum_{t=t_j+1}^{t_{j+1}} (-\log P(x_t | x^{t-1} : M_t)) \right\}, \quad (13)$$

where  $t_0 = 0 < t_1 < \dots < t_m < t_{m+1} = n$  denote change-points,  $m$  is the number of change-points,  $M(j) \in \mathcal{M}$  is the model at  $[t_j, t_{j+1})$  ( $i = 0, \dots, m$ ), and the minimum is taken under the condition that  $|M(j) - M(j+1)| \leq 1$  ( $j = 0, \dots, m-1$ ).  $F(n, m)$  is code-length for a model sequence  $M(0)..M(0)M(1)..M(m)$ . For SM estimator, KT estimator, and Westimator, we denote  $F(n, m)$  as  $F_{SM}(n, m)$ ,  $F_{KT}(n, m)$ , and  $F_W(n, m)$ , respectively. They are expanded as follows:

$$F_{SM}(n, m) = m \log \frac{n}{m} + \epsilon(m+1) \log \frac{n}{m+1} + (m+1) \log(1+\epsilon) - m \log \frac{\epsilon}{2},$$

$$F_{KT}(n, m) = (n-1)H\left(\frac{m}{n-1}\right) + \frac{1}{2} \log(n-1) + (m+1) \log 2,$$

$$F_W(n, m) = \frac{3m}{2} \log \frac{n}{m} + \frac{1}{2} \log n + (2m-1) \log 2 + m,$$

where  $H(x) = -x \log x - (1-x) \log(1-x)$ .

For each  $m$ , for any sufficiently large  $n$ , for sufficiently small  $\epsilon > 0$ , the following relation holds among SM, KT, and W:

$$F_{SM}(n, m) < F_{KT}(n, m) < F_W(n, m). \quad (14)$$

### 4.2. Learning PSMSs

Let  $\mathcal{X}$  be either discrete or continuous. Let  $\mathcal{F} = \{p(x; \theta) : \theta \in \Theta\}$  be a parametric class of probability distributions (or probability mass functions) where  $\Theta$  is a parameter space. We suppose that each  $x_t$  of  $x^n = x_1 \dots x_n \in \mathcal{X}^n$  is independently generated according to a class of probability distributions with  $m+1$  piecewise constant parameters as follows:

$$\begin{cases} x_t \sim p(x; \theta(0)) & (1 \leq t \leq t_1), \\ x_t \sim p(x; \theta(1)) & (t_1 + 1 \leq t \leq t_2), \\ \vdots \\ x_t \sim p(x; \theta(m)) & (t_m + 1 \leq t \leq n), \end{cases} \quad (15)$$

where  $0 < t_1 < t_2 < \dots < t_m < n$  ( $t_0 = 0, t_{m+1} = n$ ) is a sequence of change-points and each  $\theta(j) \in \Theta$  ( $j = 0, \dots, m$ ) and  $\theta(j) \neq \theta(j+1)$  ( $j = 0, \dots, m-1$ ). We call such a source a *piecewise stationary memoryless source* (PSMS) [7],[9].

We consider any lossless data compression algorithm  $\mathcal{A}$ , which takes as input  $x^n$  and outputs a lossless compressed data sequence. We denote the total code-length for  $x^n$  using  $\mathcal{A}$  as  $\mathcal{L}_{\mathcal{A}}(x^n)$ . We define as a measure for the goodness of  $\mathcal{A}$  the *expected redundancy* as follows:

**Definition 11** For any lossless data compression algorithm  $\mathcal{A}$ , for a given PSMS as in (15), we define the *expected redundancy* for  $\mathcal{A}$  by

$$\mathcal{R}_{\mathcal{A}}^n \stackrel{\text{def}}{=} \mathbb{E} \left[ \mathcal{L}(x^n) - \sum_{j=0}^m \sum_{t=t_j+1}^{t_{j+1}} (-\log p(x_t; \theta(j))) \right],$$

where the expectation is taken with respect to (15).

Merhav[7] derived the following lower bound on the expected redundancy.

**Theorem 12** [7] *Suppose that the domain  $\mathcal{X}$  is finite. Supposing that each datum is independently generated according to almost any PSMS with fixed  $m$  as the number of change-points and fixed  $k$  as the degrees of freedom of each parameter, and under other some conditions for any  $\epsilon > 0$  and sufficiently large  $n$ , we have*

$$\inf_{\mathcal{A}} \mathcal{R}_{\mathcal{A}}^n \geq (1-\epsilon) \left( \frac{k(m+1)}{2} \log n + m \log n \right). \quad (16)$$

In the case where  $\Theta$  is 1-dimensional and compact, Kanazawa and Yamanishi[4] applied DMS to develop an algorithm that asymptotically matched (16). Below we introduce their approach. The key ideas of their algorithm are summarized as follows:

1) *Discretization of parameter space:* For a given positive integer  $K$ , we discretize  $\Theta$  to obtain a finite set of size  $K$ . Let us define Fisher information associated with  $\mathcal{F}$  and  $L_I$  by

$$I(\theta) \stackrel{\text{def}}{=} \mathbb{E}_{\theta} \left[ -\frac{\partial^2 \log p(x; \theta)}{\partial \theta^2} \right], \quad L_I \stackrel{\text{def}}{=} \int_{\theta \in \Theta} \sqrt{I(\theta)} d\theta,$$

respectively. Letting  $\delta_I = L_I / (K-1)$  be a discretization scale and  $\bar{\theta}_1 = \theta_{\min}$ , we define  $\bar{\theta}_i$  so that

$$\int_{\bar{\theta}_1}^{\bar{\theta}_i} \sqrt{I(\theta)} d\theta = (i-1) \delta_I \quad (i = 2, \dots, K). \quad (17)$$



We have  $\bar{\Theta} = \{\bar{\theta}_1, \dots, \bar{\theta}_K\}$ . We assume that for each interval  $\bar{\theta}_i \leq \theta \leq \bar{\theta}_{i+1}$ , either  $d\sqrt{I(\theta)}/d\theta \leq 0$  or  $d\sqrt{I(\theta)}/d\theta \geq 0$ .

2) *Settings of model transition probabilities:* When the model set is a set of discretized parameters, it may be difficult to assume that the parameter transits to neighbouring ones only as in Definition 1). In that case, we assume according to [4] that the parameter value transits according to the following probabilities:

$$\Pr(i_t | i_{t-1}) = \begin{cases} \frac{\alpha}{K-1} & (i_t \neq i_{t-1}), \\ 1 - \alpha & (i_t = i_{t-1}). \end{cases} \quad (18)$$

where we set  $K$  and  $\alpha$  as

$$K = \lfloor \sqrt{n} \rfloor, \quad \alpha = 1/n.$$

Under the above setting Kanazawa and Yamanishi [4] proposed an algorithm for learning PSMSs that takes  $x^n$  as input and outputs the parameter sequence  $(\bar{\theta}_{i_1}, \dots, \bar{\theta}_{i_n})$  where  $i_1, \dots, i_n$  are those which attain the DMS criterion. Its performance is summarized in the following theorem:

**Theorem 13** [4] *Suppose that each datum is independently drawn according to a PSMS. There exists an algorithm  $\mathcal{A}$  for which time complexity is  $O(n^{3/2})$  and the expected redundancy satisfies:*

$$\mathcal{R}_{\mathcal{A}}^n < \frac{m+1}{2} \log n + m \log n + \frac{L_I^2}{2} + \log e + O(n^{-1/2}). \quad (19)$$

The bound (19) implies that the expected redundancy for the algorithm asymptotically matches the lower bound (16).

## 5. DISTINGUISHABILITY

Let us employ  $\mathcal{F} = \{p(x; \theta) : \theta \in \Theta\}$  as a model class of probability distributions (or probability mass functions) where  $\Theta$  is a 1-dimensional real-valued parameter space. We consider how to discretize  $\Theta$  to get a finite subset  $\bar{\Theta}$ . From the argument in Section 4.2 (see 17), we see that if we let the discretization scale  $\delta = \max_i |\bar{\theta}_i - \bar{\theta}_{i+1}|$  be

$$\delta = O\left(\sqrt{1/n}\right) \quad (20)$$

then we have an upper bound on the expected redundancy which attains Merhav's lower bound. In this sense the discretization scale as in (20) is optimal in the scenario of data compression. It coincides with results in [8],[1].

Meanwhile, let us consider the case where DMS is applied into change-point detection over a discretized parameter set  $\bar{\Theta}$ . When either SM, KT, W estimator or the uniform model transition probability as in (18) is employed for model transition estimation, we see from Theorem 8 that Type 2 error probability for DMS decreases exponentially with respect to  $n$  if

$$\min_{\bar{\theta}(\neq)\bar{\theta}' \in \bar{\Theta}} D(\bar{\theta}||\bar{\theta}') > f(n, t^*)/n = O(\log n/n). \quad (21)$$

Note that for any  $\bar{\theta}, \bar{\theta}' \in \bar{\Theta}$ , we have  $D(\bar{\theta}||\bar{\theta}') = (1/2)I(\bar{\theta})\delta^2$ , where  $\delta$  is the discretization scale. If

$$\delta = O\left(\sqrt{\log n/n}\right) \quad (22)$$

then (21) holds. The discretization scale (22) makes the total code-length  $(1/2) \log n$  larger than the bound (19). This implies that (22) doesn't lead to optimal data compression. Hence there is a gap between the optimal discretization in the sense of change-detection and that of data compression. Change-detection requires more discriminability over the parameter space than data compression.

## 6. CONCLUSION

We have applied DMS into the scenarios of change-detection and data compression for time-varying sources. We have analyzed the performance of DMS in the both scenarios and have shown how it is related to model transition estimation. We have argued how to discretize the real-valued parameter space to obtain optimal performance in the both scenarios. It has turned out that change-detection may require more discriminability over the parameter space than data compression.

## 7. ACKNOWLEDGMENTS

This work was partially supported by MEXT KAKENHI 23240019, Aihara Project, the FIRST program from JSPS, initiated by CSTP, NTT Corporation.

## 8. REFERENCES

- [1] V. Blasiarmanian. Statistical inference, Occam's razor and statistical mechanics on the space of probability distributions. *Neural Computation*, 9, No.2, pp:349–368, 1997.
- [2] T. van Erven and P.D. Grünwald and S. de Rooij. Catching up faster in Bayesian model selection and model averaging. *Advances in NIPS 20*, 2007.
- [3] S. Hirai and K. Yamanishi. Detecting changes of clustering structures using normalized maximum likelihood coding. *Proc. of the eighteenth ACM SIGKDD Int'l. Conf. on Knowledge Discovery in Data Mining (KDD2012)*, 2012.
- [4] H. Kanazawa and K. Yamanishi. An MDL-based change-detection with its applications to learning piecewise stationary memoryless sources. *Proc. of IEEE Information Theory Workshop (ITW2012)*, 2012.
- [5] J. Kleinberg. Bursty and hierarchical structure in streams. *D. M. K. D.*, vol. 7, pp. 373–397, Nov. 2003.
- [6] R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Trans. Inf. Theory*, 27:199–207, 1981.
- [7] N. Merhav. On the minimum description length principle for sources with piecewise constant parameters. *IEEE Trans. Inf. Theory*, vol. 39, pp. 1962–1967, Nov. 1993.
- [8] J. Rissanen. *Information and Complexity in Statistical Modeling*, Springer, 2007.
- [9] G. I. Shamir and D. J. Costello, Jr. Asymptotically optimal low-complexity sequential lossless coding for piecewise-stationary memoryless sources—Part I: The regular case. *IEEE Trans. Inf. Theory*, vol. 46, pp. 2444–2467, 2000.
- [10] G. I. Shamir and N. Merhav. Low complexity sequential lossless coding for piecewise stationary memoryless sources. *IEEE Trans. Inf. Theory*, Vol.45, pp:1498–1519, 1999.
- [11] F. M. J. Willems. Coding for a binary independent piecewise identically-distributed source. *IEEE Trans. Inf. Theory*, Vol.42, pp:2210–2217, 1996.
- [12] F. M. J. Willems and F. Casadei. Weighted coding methods for binary piecewise memoryless sources. *Proc. of 1995 IEEE ISIT*, p.323, 1995.
- [13] K. Yamanishi and Y. Maruyama. Dynamic syslog mining for network failure monitoring. *Proc. of KDD2005*, pp: 499–508, ACM Press, 2005.
- [14] K. Yamanishi and Y. Maruyama. Dynamic model selection with its applications to novelty detection. *IEEE Trans. Inf. Theory*, IT 53(6) : 2180–2189, 2007.

# CLUSTERING CHANGE DETECTION USING NORMALIZED MAXIMUM LIKELIHOOD CODING

So Hirai<sup>1</sup>, Kenji Yamanishi<sup>2</sup>

<sup>1</sup>Graduate School of Information Science and Engineering, The University of Tokyo,  
7-3-1, Hongo, Bunkyo-ku, Tokyo, JAPAN,

(Currently belonging to NTT DATA Corporation.) so.hiral.16@gmail.com,

<sup>2</sup>Graduate School of Information Science and Engineering, The University of Tokyo,  
7-3-1, Hongo, Bunkyo-ku, Tokyo, JAPAN, yamanishi@mist.i.u-tokyo.ac.jp

## ABSTRACT

We are concerned with the issue of detecting changes of clustering structures from multivariate time series. From the viewpoint of the minimum description length (MDL) principle, we introduce an algorithm that tracks changes of clustering structures so that the sum of the code-length for data and that for clustering changes is minimum. Here we employ a Gaussian mixture model (GMM) as representation of clustering, and compute the code-length for data sequences using the normalized maximum likelihood (NML) coding. The introduced algorithm enables us to deal with clustering dynamics including merging, splitting, emergence, disappearance of clusters from a unifying view of the MDL principle. We empirically demonstrate using artificial data sets that our proposed method is able to detect cluster changes significantly more accurately than an existing statistical-test based method and AIC/BIC-based methods. We further use real customers' transaction data sets to demonstrate the validity of our algorithm in market analysis.

## 1. SUMMARY

### 1.1. Problem Setting

This paper is organized as a brief summary of our recent paper [1]. We address the issue of clustering multi-variate data sequences. Suppose that the nature of data changes over time. We are then specifically interested in tracking changes of clustering structures, which we call *clustering change detection*. We are concerned with the situation where time series data are sequentially given and the clustering must be conducted in a sequential fashion. The main purpose of this talk is to introduce, according to our recent work [1], a novel clustering change detection algorithm in the sequential setting. We employ a Gaussian mixture model (GMM) as a representation of clustering and design the algorithm on the basis of the minimum description length (MDL) principle [2]. That is, it tracks changes of clustering structures so that the sum of the code-length for data and that for clustering changes is minimum.

### 1.2. Previous Works

There exist a number of methods for tracking changes of clustering structures. For example, Song and Wang [3] proposed a statistical-test based algorithm for dynamic clustering. It estimates a GMM in an on-line manner and then conducts a statistical test to determine whether a new cluster is identical to an old one or not. If it is, the new cluster is merged into the older one, otherwise it is recognized as a cluster which has newly emerged. Sato [4] proposed an algorithm for merging and splitting of clusters in a GMM based on the variational Bayes method. Note that changes of clusters are not necessarily classified into merging or splitting. Siddiqui et.al.[5] proposed a method of tracking clustering changes using the EM algorithm and Kalman filters. Our work is different from Siddiqui et.al.'s one in that the former is concerned with changes of the number of clusters while the latter is concerned with parameter trajectories keeping the number of clusters fixed.

### 1.3. Novelty of Our Approach

The novelty of the approach in [1] may be summarized as follows:

1) *An extension of DMS into a sequential clustering setting*: Yamanishi and Maruyama [6, 7] developed a theory of dynamic model selection (DMS) for tracking changes of statistical models on the basis of the MDL principle. We extend DMS to the sequential setting to introduce a *sequential DMS algorithm* [1]. Every time data is input, it sequentially detects changes of clustering structures on the basis of the MDL principle so that the sum of the code-length for the data and that for the clustering change is minimum. This algorithm enables us to deal with the dynamics of clustering structures, including “merging”, “splitting”, “emergence”, “disappearance”, etc. within a unified framework from the viewpoint of the MDL principle.

2) *A new application of the NML code-length to sequential DMS*: In the sequential DMS algorithm, it is crucial how to choose a method for coding. The best choice is the NML coding since it has turned out to be the optimal

code-length in the sense of minimax criterion [2]. However, the normalization term diverges for a multi-dimensional Gaussian distribution and it is computationally difficult to straightforwardly compute the NML code-length for a GMM exactly. Hirai and Yamanishi proposed a method for efficiently computing the NML code-length for GMMs [8], inspired by Kontkanen and Myllymäki's work [9] in which the efficient computation of the NML code-lengths for discrete distributions was addressed. They recently modified their method using the renormalizing technique as in [10], to develop an efficient method for computing the renormalized maximum likelihood code-length (RNML) for a GMM [11]. We employ the RNML coding for GMMs in the computation process of the sequential DMS. This is the first work on the usage of the RNML coding in the scenario of sequential clustering change detection.

3) *Empirical demonstration of the superiority of the sequential DMS with the RNML code-length over the existing methods:* Using artificial data sets, we empirically demonstrate the validity of our method in comparison with Song and Wang's method [3], AIC (Akaike's information criteria)[12]/BIC (Bayesian information criteria)[13]-based tracking methods etc. We also use a real data set consisting of customers' purchase records for a number of kinds of beers. Tracking changes of clusters of customers leads to the understanding of how customers' purchase patterns change over time and how customers move from clusters to clusters. This demonstrates the validity of our method in the area of marketing.

## 2. ACKNOWLEDGMENTS

This work was supported by MEXT KAKENHI 23240019, Aihara Project, the FIRST program from JSPS, initiated by CSTP, NTT Corporation.

## 3. REFERENCES

- [1] S.Hirai and K. Yamanishi, "Detecting changes of clustering structures using normalized maximum likelihood coding," *Proc. of KDD2012*, 2012.
- [2] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. on Inf. Theory*, vol. 42(1), pp. 40–47, January 1996.
- [3] M. Song and H. Wang, "Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering," *Intelligent Computing: Theory and Application*, 2005.
- [4] M. Sato, "Online model selection based on the variational bayes," *NC*, vol. 13, pp. 1649–1681, 2001.
- [5] Z.F.Siddiqui G.Kreml and M.Spiliopoulou, "Online clustering of high-dimensional trajectories under concept drift," *Proc. of ECML-PKDD2011, Part II*, pp. 261–276, 2011.
- [6] K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," *Proc. of KDD2005*, pp. 499–508, 2005.
- [7] K. Yamanishi and Y. Maruyama, "Dynamic model selection with its applications to novelty detection," *IEEE Trans. on Inf. Theory*, vol. 53, no. 6, pp. 2180–2189, June 2007.
- [8] S. Hirai and K. Yamanishi, "Normalized maximum likelihood coding for exponential family with its applications to optimal clustering," *arXiv 0474364*, 2012.
- [9] P. Kontkanen and P. Myllymäki, "A linear time algorithm for computing the multinomial stochastic complexity," *Information Processing Letters*, vol. 103, pp. 227–233, 2007.
- [10] J. Rissanen, "MDL denoising," *IEEE Transactions on Information Theory*, vol. 46, no. 7, pp. 2537–2543, November 2000.
- [11] S. Hirai and K. Yamanishi, "Efficient computation of normalized maximum likelihood coding for Gaussian mixtures with its applications to optimal clustering," *Proc. of ISIT*, pp. 1031–1035, 2011.
- [12] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.
- [13] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics* 6 (2), pp. 461–464, 1978.



# COMPARISON OF NML AND BAYESIAN SCORING CRITERIA FOR LEARNING PARSIMONIOUS MARKOV MODELS

Ralf Eggeling<sup>1</sup>, Teemu Roos<sup>2</sup>, Petri Myllymäki<sup>2</sup>, Ivo Grosse<sup>1</sup>

<sup>1</sup>Institute for Computer Science, Martin Luther University Halle-Wittenberg, 06099 Halle, GERMANY, {eggeling|grosse}@informatik.uni-halle.de

<sup>2</sup>Helsinki Institute for Information Technology HIIT, University of Helsinki, P.O.Box 68, FIN-00014 Helsinki, FINLAND, {teemu.roos|petri.myllymaki}@hiit.fi

## ABSTRACT

Parsimonious Markov models, a generalization of variable order Markov models, have been recently introduced for modeling biological sequences. Up to now, they have been learned by Bayesian approaches. However, there is not always sufficient prior knowledge available and a fully uninformative prior is difficult to define. In order to avoid cumbersome cross validation procedures for obtaining the optimal prior choice, we here adapt scoring criteria for Bayesian networks that approximate the Normalized Maximum Likelihood (NML) to parsimonious Markov models. We empirically compare their performance with the Bayesian approach by classifying splice sites, an important problem from computational biology.

## 1. INTRODUCTION

Classifying discrete sequences is an omnipresent task in computational biology, where an additional challenge is limited data. Recently, parsimonious Markov models [1], a generalization of variable order Markov models [2], have been proposed to model complex statistical dependencies among adjacent observations while keeping the parameter space small and thus avoiding overfitting.

Parsimonious Markov models (parsMMs) use parsimonious context trees (PCTs), which differ from traditional context trees [2] in two aspects: (i) a PCT is a balanced tree, i.e. each leaf has the same depth, and (ii) each node represents an arbitrary subset of the alphabet  $\mathcal{A}$ , with the additional constraint that everywhere in the tree, sibling nodes form together a partition of  $\mathcal{A}$ . An example PCT, which shows both features, forming a partition of context sequences that can not be represented by a traditional context tree, is shown in Figure 1. A PCT  $\tau$  of depth  $d$  partitions all *context sequences* of length  $d$  over alphabet  $\mathcal{A}$  into disjoint sets, which are called *context*. We denote all contexts represented by  $\tau$  as  $\mathcal{C}_\tau$ . An inhomogeneous parsimonious Markov model of order  $D$  for modelling sequences of length  $L$  allows using different PCTs at each position in the sequence. The first  $D$  positions use PCTs of increasing order  $0, \dots, D - 1$ , whereas the remaining  $L - D$  positions use PCTs of order  $D$ . The likelihood

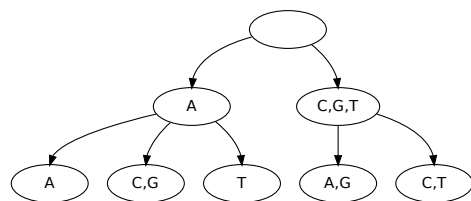


Figure 1. Example PCT of depth 2 over DNA alphabet. It encodes the partitioning of all 16 possible sequences of length 2 into a set of contexts  $\mathcal{C}_\tau = \{\{AA\}, \{CA, GA\}, \{TA\}, \{AC, AG, AT, GC, GG, GT\}, \{CC, CG, CT, TC, TG, TT\}\}$ .

function is given by

$$P(\mathbf{X}|\vec{\Theta}) = \prod_{\ell=1}^L \prod_{\mathbf{w} \in \mathcal{C}_{\tau_\ell}} \prod_{a \in \mathcal{A}} (\theta_{\ell \mathbf{w} a}^{\tau_\ell})^{N_{\ell \mathbf{w} a}}. \quad (1)$$

where  $N_{\ell \mathbf{w} a}$  is the number of occurrences of symbol  $a$  at position  $\ell$  in all sequences in data set  $\mathbf{X}$ , whose subsequences from position  $\ell - |\mathbf{w}|$  to  $\ell - 1$  are an element of context  $\mathbf{w}$ .

The likelihood is closely related to that of Bayesian networks (BNs), since it factorizes into independent terms for each variable and the number of conditional probability parameters depends on the structure of the model. However, whereas BNs have freedom in choosing the parent nodes of a random variable but always use separate conditional probability parameters for each possible realization of the parent nodes, parsMMs have fixed parent nodes but freedom in lumping several of their possible realizations together as one context.

There is an efficient dynamic programming (DP) algorithm [3, 1] for finding the PCT that maximizes an arbitrary structure score, which only has to fulfil the property of factorizing into independent leaf scores. In the Bayesian setting, the structure score is usually the local posterior probability of a PCT given data. If the local parameter prior is a symmetric Dirichlet with equivalent sample size (ESS)  $\alpha$ , we obtain the BDeu score [4], which

can be used in the DP algorithm since it factorizes along contexts. The conditional probability parameters are estimated by the mean posterior (MP) principle.

In practice, there is rarely reliable a priori knowledge available for specifying  $\alpha$ . Since it is known that the choice of  $\alpha$  influences the model complexity in the case of Bayesian networks [5], it is safe to assume that a similar effect may be observed for parsimonious Markov models. Often a cross validation (CV) on the training data is used to obtain a reasonable choice for this external parameter. However, CV is a time consuming procedure and there is no guarantee that a useful prior on a subset of the training data will also yield optimal results when learning from the complete training data for classifying previously unseen test instances.

In order to avoid CV, we propose using NML approximating methods for structure and parameter learning, which have been initially proposed for BNs, for parsimonious Markov models. The fNML score [6] has been suggested as score for structure learning of BNs, whereas the corresponding conditional probability parameters have been obtained in the same setting by using fsNML estimates [7]. Due to the structural similarity of the likelihood function of parsMMs and that of BNs, both methods can be adapted without modification.

## 2. RESULTS

We compare two different scores for the PCT structures, BDeu and fNML, and two different methods for estimating conditional probability parameters of each PCT, MP and fsNML. In order to determine whether structure or parameter learning is dominating the results, we do not only compare MP parameter estimates for a BDeu optimal structure with fsNML parameter estimates for an fNML optimal structure, but also consider the other two possibilities (Table 1).

We perform two separate case studies. The first study is a standard classification experiment for short symbolic sequences, which uses labeled training data and involves structure and parameter learning for both classes. In computational biology, this an abundant task, when experimentally verified training data is available.

The second study is inspired by the computational problem of de novo motif discovery [8, 9]. Motif discovery usually involves latent variables, hence it cannot be solved exactly, and approximate algorithms, such as the expectation-maximization (EM) algorithm [10] have to be resorted to. Formulating fNML and fsNML in a setting with latent variables, i.e. utilizing weighted data inside the EM algorithm is not straightforward, but a slight modification of the classification problem resembles the task that typically arises in those iterative algorithms. In the modified classification, the structure and parameters of the background class are fixed and there is much more background training data available. Hence the prior in the Bayesian setting only affects the foreground model. This resembles the problem of motif discovery, where only structure and parameters of a motif model (foreground)

Table 1. The two combinations in the major diagonal are the obvious ways of learning parsMMs in the Bayesian and NML setting respectively, whereas the minor diagonal contains rather artificial combinations, which we mainly investigate for academic purposes.

	BDeu	fNML
MP	BDeu-MP	fNML-MP
fsNML	BDeu-fsNML	fNML-fsNML

are to be estimated, whereas the structure and parameters of the background model remain fixed.

### 2.1. Standard classification

In the first experiment, we perform a standard classification on the benchmark data set of Yeo and Burge [11]. It consists of 12,623 experimentally verified splice donor sites (foreground data) and 269,157 non splice sites (background data). Both data sets, consisting of sequences of length 7 over the quaternary DNA alphabet, were already split by Yeo and Burge into training and test data at the ratio of 2:1 [11], and we use the same partitioning.

Since we are interested in situations with limited data, we randomly pick 500 sequences from each of the training data sets for learning foreground and background model, both being second order inhomogeneous parsimonious Markov models. We learn – for each possible combination of scores – structure and parameters of two parsimonious Markov models. For the Bayesian scores, we learn models for a large variety of possible ESS values, ranging from  $10^{-5}$  to  $10^8$ . We repeat the procedure  $10^3$  times with different training samples.

In Figure 2, we compare the average complexities of the learned models. For the BDeu score, we observe with increasing ESS an increase in model complexity, which is a behaviour that is already known from Bayesian networks [5]. The fNML score has the advantage of not being affected by the ESS at all. However, it yields a comparatively low model complexity for the foreground model, which is surprising since the foreground data set is known to contain strong statistical dependencies. The background model is surprisingly complex, given the fact that the background data shows much less dependencies.

Additional studies have shown that the difference in model complexity of fNML estimated foreground and background model decreases when both samples sizes are reduced. The BDeu score, however, retains a certain difference in model complexity, even when sample sizes are very small.

However, the PCT structure itself is not sufficient to compare scoring criteria, since we are mainly interested in the classification performance of the learned models. In order to evaluate the classification performance of a set of PCTs, we estimate conditional probability parameters, build a likelihood ratio classifier, compute probabilities for each sequence in both test data sets and compute

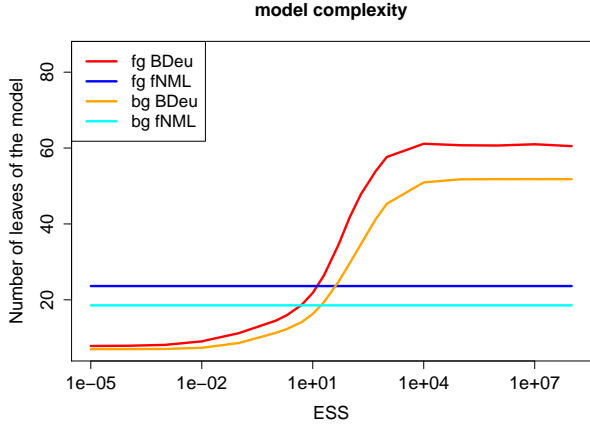


Figure 2. Averaged model complexities (measured as the total number of leaves in the model) for foreground and background model are plotted against the equivalent sample size. Since the fNML criterion does not use the ESS parameter, the model complexities is constant. Standard errors are 0.1 at most, hence error bars are omitted from the plot.

the area under the ROC curve (AUC) [12]. When combining the Bayesian structure and parameter learning, we apply the same prior to both problems.

For each of the four possible score combinations, we repeat the entire study with  $10^3$  different training samples and average the resulting AUC values. The results are shown in Figure 3. We observe an AUC of 0.9691 for the fNML-fsNML method. For an ESS ranging from  $10^1$  to  $10^3$ , the Bayesian approach outperforms fNML-fsNML method, obtaining a maximal AUC of 0.9708 for an ESS of 200. Interestingly, an ESS of 1, which is often considered to be the most uninformative choice, is obviously not optimal, since performs significantly worse than larger ESS values and even slightly worse than the NML approach.

The mixed approach of combining fNML structure learning with MP parameter estimates also yields a good classification, if the ESS is chosen correctly. For ESS values between 10 and 500, it outperforms the pure NML method, and its absolute maximum with an AUC of 0.9712 at ESS of 100 even outperforms the pure Bayesian method, even though the difference is quite small.

The BDeu-fsNML method does not show strong over- or underfitting, but it is even with perfectly chosen ESS only slightly better than the pure NML method. In general, the parameter learning seems to dominate the experiment, since the methods using the same parameter estimate resemble each other more than the methods using the same structure score.

## 2.2. Fixed background

In the second experiment, we consider a different setting. Now fix the background model to a simple independence model and estimate its parameters once from

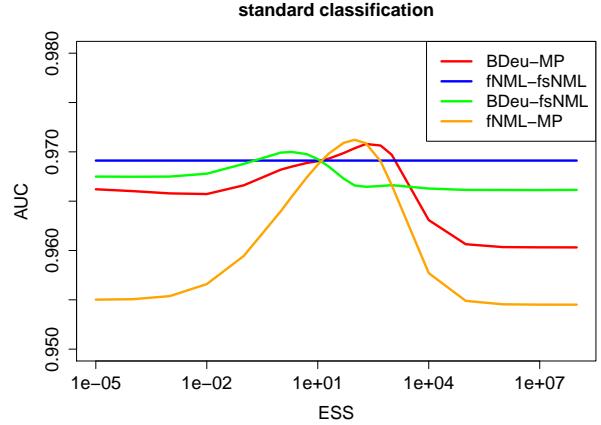


Figure 3. Averaged AUC values for the standard classification experiment plotted against the equivalent sample size. In the BDeu-MP setting, the same ESS is used for structure and parameter learning. For BDeu-fsNML, the ESS only affects structure learning, whereas for fNML-MP is only affects parameter learning. Standard errors are  $10^{-4}$  at most, hence error bars are omitted.

the entire background training data set according to the maximum likelihood (ML) principle. Since the complete background data contains over  $10^5$  data points, the ML estimator is basically identical to fsNML and MP estimates. The repeated holdout experiment as described in the previous section is only carried out for the foreground model. This situation resembles the problem de novo motif discovery [8, 9], where there is orders of magnitude more data available for learning the parameters of the background compared to the foreground, and where learning the background model does not contain a model selection step.

The results of this modified classification are shown in Figure 4. We observe the fNML-fsNML approach in comparison with the BDeu-MP approach to be almost optimal. There is only a tiny improvement that the Bayesian approach may achieve if the ESS would have been chosen perfectly at a value of approximately 20. Interestingly, both mixed approaches perform better than the pure Bayesian approach, since the range of good ESS values and the maximal improvement in AUC are increased.

Both methods using the MP parameter estimates break down if the ESS is larger than 100, which might be explained as follows. If the foreground parameters are computed by using a large ESS, resulting large pseudocounts, they get concentrated around the uniform distribution. This is not a problem as long as the same applies to the background parameters, since even small differences between foreground and background parameters are sufficient to classify a test sequence correctly. However, if the background parameters are fixed to certain values, only smoothing the foreground parameters creates an imbalance which prevents a fair comparison of foreground and background likelihood for a test sequence, resulting in

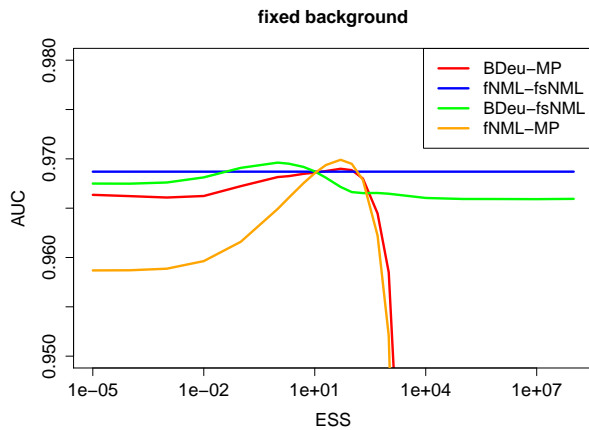


Figure 4. Averaged AUC values for the classification experiment with fixed background model. The standard errors are below  $10^{-5}$ , hence error bars are omitted.

many classification errors. This situation however, typically occurs in the problem of de novo motif discovery, where a motif model is estimated from small data samples, and where the background model, it is compared with, has fixed parameters that may have been estimated from a much larger amount of data.

### 3. CONCLUSIONS

We have compared NML with Bayesian criteria for structure and parameter learning of parsimonious Markov models with application to the classification of DNA sequences. In a standard classification, we found the Bayesian approach to perform well, outperforming the NML approach for a comparatively large range of ESS values. We also found the optimal ESS parameter for classification purposes to be larger than 1, which is often an intuitive choice, but smaller than 500. In a classification with fixed background model structure and parameters, we found the NML approach to be as good as the optimal Bayesian approach. The latter does not yield a significant improvement in AUC, even if the optimal value of the ESS would have been guessed. Moreover, we find the Bayesian approach in this setting to be very sensitive towards very large ESS values. This makes it tempting to speculate that the NML learning approach might be also of use in the problem of de novo motif discovery, which includes a classification step with fixed background parameters.

### 4. ACKNOWLEDGMENTS

This work was funded by *Reisestipendium des allg. Stiftungsfonds der MLU Halle-Wittenberg* and the Academy of Finland (PRIME and MODEST).

### 5. REFERENCES

[1] Pierre-Yves Bourguignon, *Parcimonie dans les modèles markoviens et applications à l'analyse des*

*séquences biologiques*, Ph.D. thesis, Université Evry Val d'Essonne, 2008.

- [2] Jorma Rissanen, “A universal data compression system,” *IEEE Trans. Inform. Theory*, vol. 29, no. 5, pp. 656–664, 1983.
- [3] P. Bühlmann and A.J. Wyner, “Variable length Markov chains,” *Annals of Statistics*, vol. 27, pp. 480–513, 1999.
- [4] G. Heckerman, D. Geiger, and D. Chickering, “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data,” *Machine Learning*, vol. 20, pp. 197–243, 1995.
- [5] T. Silander, P. Kontkanen, and P. Myllymäki, “On Sensitivity of the MAP Bayesian Network Structure to the Equivalent Sample Size Parameter,” in *Proceedings of the The 23rd Conference on Uncertainty in Artificial Intelligence (UAI-2007)*, 2007, pp. 360–367.
- [6] T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki, “Factorized NML Criterion for Learning Bayesian Network Structures,” in *Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM-08)*, 2008.
- [7] T. Silander, T. Roos, and P. Myllymäki, “Locally Minimax Optimal Predictive Modeling with Bayesian Networks,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009, pp. 504–511.
- [8] C.E. Lawrence and A.A. Reilly, “An Expectation Maximization Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences,” *Proteins: Structure, Function and Genetics*, vol. 7, pp. 41–51, 1990.
- [9] T.L. Bailey and C. Elkan, “Fitting a mixture model by expectation maximization to discover motifs in biopolymers,” in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 1994, pp. 28–36.
- [10] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
- [11] G. Yeo and C.B. Burge, “Maximum Entropy Modeling of Short Sequence Motifs with Applications to RNA Splicing Signals,” *Journal of Computational Biology*, vol. 11(2/3), pp. 377–394, 2004.
- [12] Kent A. Spackman, “Signal detection theory: Valuable tools for evaluating inductive learning,” in *Proceedings of the Sixth International Workshop on Machine Learning*, San Mateo, CA, 1989, pp. 160–163.

# CONVEX FORMULATION FOR NONPARAMETRIC ESTIMATION OF MIXING DISTRIBUTION

Kazuho Watanabe<sup>1</sup> and Shiro Ikeda<sup>2</sup>

<sup>1</sup>Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5, Takayama-cho, Ikoma-shi, Nara, 630-0192, JAPAN, wkazuho@is.naist.jp

<sup>2</sup>The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa-shi, Tokyo, 190-8562 JAPAN, shiro@ism.ac.jp

## ABSTRACT

We discuss a nonparametric estimation method of the mixing distribution in mixture models. We propose an objective function with one parameter, where its minimization becomes the maximum likelihood estimation or the kernel vector quantization in special cases. Generalizing Lindsay's theorem for the nonparametric maximum likelihood estimation, we prove the existence and discreteness of the optimal mixing distribution and devise an algorithm to calculate it. Furthermore, we show the connection between the unifying estimation framework and the rate-distortion problem. It is demonstrated that with an appropriate choice of the parameter, the proposed method is less prone to overfitting than the maximum likelihood method.

## 1. INTRODUCTION

Mixture models are widely used for clustering and density estimation. We discuss a nonparametric estimation method of mixture models where an arbitrary distribution, including a continuous one, is assumed over the component parameter. It was proved by Lindsay [1] that the maximum likelihood estimate of the mixing distribution is given by a discrete distribution whose support consists of distinct points, the number of which is no more than the sample size. This provides a framework for determining the number of mixture components from data. The mixture estimation algorithm developed in [2] can be considered as a procedure for estimating such discrete distributions. However, it is vulnerable to overfitting because of the flexibility of the nonparametric estimation.

In this study, we propose a nonparametric mixture estimation method defined by minimization of an objective function with one parameter  $\beta$ . With specific choices of  $\beta$ , the proposed method reduces to the maximum likelihood estimation (MLE) and the kernel vector quantization (KVQ) [3]. Generalizing Lindsay's theorem for the nonparametric MLE, we prove the existence and discreteness of the optimal mixing distribution. Then, we provide an algorithm to calculate the optimal discrete distribution, that is specifically tailored to the proposed objective function from the procedure in [2]. Numerical experiments demonstrate that there exists an appropriate choice of  $\beta$

in terms of the average generalization error. Furthermore, we relate the proposed mixture estimation method to the rate-distortion problem [4] to build insight into the selection of the width of the component density.

## 2. MIXTURE MODELLING

Given  $n$  training samples,  $\{x_1, \dots, x_n\}$ ,  $x_i \in R^d$ , consider nonparametric estimation of the mixing distribution  $q(\theta)$  of the following mixture density of the model  $p(x|\theta)$  with parameter  $\theta \in \Omega$ ,

$$r(x) = r(x; q) = \int p(x|\theta)q(\theta)d\theta. \quad (1)$$

Let  $r_i = r(x_i; q) = \int p(x_i|\theta)q(\theta)d\theta$ . We choose  $q(\theta)$  as the optimal function of the following problem,

$$\hat{q}(\theta) = \underset{q}{\operatorname{argmin}} F_\beta(q),$$

where

$$F_\beta(q) = \begin{cases} \frac{1}{\beta} \log \left( \frac{1}{n} \sum_{i=1}^n r_i^{-\beta} \right), & (\beta \neq 0) \\ -\frac{1}{n} \sum_{i=1}^n \log r_i & (\beta = 0). \end{cases} \quad (2)$$

The objective function  $F_\beta(q)$  is continuous with respect to  $\beta \in R$ . This estimation boils down to the MLE when  $\beta = 0$  [1]. As  $\beta \rightarrow \infty$ , it becomes the minimization of  $\max_i(-\log r_i)$ , that is, KVQ with the kernel function,  $K(x, \theta) = p(x|\theta)$  [3]<sup>1</sup>.

For  $\beta \neq 0$ , it is also expressed as

$$F_\beta(q) = -\frac{1}{\beta} \min_{\mathbf{p} \in \Delta} \left\{ \beta \sum_{i=1}^n p_i \log r_i + \sum_{i=1}^n p_i \log \frac{p_i}{1/n} \right\}, \quad (3)$$

where  $\Delta = \{\mathbf{p} = (p_1, p_2, \dots, p_n) | p_i \geq 0, \sum_{i=1}^n p_i = 1\}$ . This expression is verified through the fact that the minimum is attained by

$$p_i = \frac{r_i^{-\beta}}{\sum_{j=1}^n r_j^{-\beta}}, \quad (4)$$

and will be used for deriving a simple learning procedure in the next section.

<sup>1</sup>The original KVQ restricts the possible support points of  $q(\theta)$  to the training data set  $\{x_1, \dots, x_n\}$ . That is  $q(\theta) = \sum_{i=1}^n q_i \delta(\theta - x_i)$ ,  $q_i \geq 0$ ,  $\sum_{i=1}^n q_i = 1$ .

### 3. OPTIMAL MIXING DISTRIBUTION

#### 3.1. Discreteness of the Optimal Mixing Distribution

We can show the convexity of  $F_\beta$  with respect to  $\mathbf{r} = (r_1, \dots, r_n)$  for  $\beta \geq -1$ .

Therefore, for  $\beta \geq -1$ , there exists a unique  $\mathbf{r}$  that minimizes  $F_\beta$  at the boundary of the convex hull of the set  $\{\mathbf{p}_\theta = (p(x_1|\theta), \dots, p(x_n|\theta)) | \theta \in \Omega\}$  where  $\Omega$  is the parameter space. From Caratheodory's theorem, this means that the optimal  $\mathbf{r}$  is expressed by a convex combination,  $\sum_{l=1}^k q_l \mathbf{p}_{\theta_l}$ , with  $q_l \geq 0$ ,  $\sum_{l=1}^k q_k = 1$  and  $k \leq n$ , indicating that the optimal mixing distribution is  $q(\theta) = \sum_{l=1}^k q_l \delta(\theta - \theta_l)$ , the discrete distribution whose support size is no more than  $n$ .

#### 3.2. Learning Algorithm

The KKT condition for the optimal  $q(\theta)$  is given by  $\mu(\theta) \leq 1$  for all  $\theta$  where

$$\mu(\theta) = \sum_{i=1}^n \alpha_i p(x_i|\theta), \quad (5)$$

and

$$\alpha_i = \frac{r_i^{-\beta-1}}{\sum_{j=1}^n r_j^{-\beta}}. \quad (6)$$

Hence the mixing distribution  $q(\theta)$  can be optimized by Algorithm 1 which sequentially augments the set of the support points until the maximum of  $\mu(\theta)$  approach 1 [2].

---

#### Algorithm 1 Decoupled Approach to Mixture Estimation

---

- 1: Initialize  $k = 0$  and  $\alpha_i = 1/n$  and prepare a small positive constant  $\epsilon$ .
  - 2: **repeat**
  - 3: Let  $\hat{\theta}_k = \operatorname{argmax}_\theta \mu(\theta)$  and  $k = k + 1$ , where  $\mu(\theta)$  is given by eq.(5).
  - 4: Define the discrete distribution,  $q_k(\theta) = \sum_{l=1}^k \pi_l \delta(\theta - \hat{\theta}_l)$ . Optimize  $\{\pi_l, \hat{\theta}_l\}_{l=1}^k$  by minimizing  $F_\beta(q_k)$ .
  - 5: Compute  $\{\alpha_i\}_{i=1}^n$  by eq.(6) with  $r_i = \sum_{l=1}^k \pi_l p(x_i|\hat{\theta}_l)$ .
  - 6: **until**  $\max_\theta \mu(\theta) < 1 + \epsilon$  holds.
- 

#### 3.3. EM Updates for Finite Mixtures

Eq.(3) is equivalent to a weighted sum of negative log-likelihood and an EM-like algorithm is available for the optimization of  $\{\pi_l, \hat{\theta}_l\}_{l=1}^k$  in Step 4. Its updating rule is obtained as follows,

$$\pi_j^{(t+1)} = \sum_{i=1}^n p_i^{(t)} \nu_{ij}, \quad \text{and} \quad \hat{\theta}_j^{(t+1)} = \frac{\sum_{i=1}^n p_i^{(t)} \nu_{ij} x_i}{\sum_{i=1}^n p_i^{(t)} \nu_{ij}},$$

where  $p_i^{(t)} = \frac{r_i^{(t)-\beta}}{\sum_{j=1}^n r_j^{(t)-\beta}}$ ,  $r_i^{(t)} = \sum_{l=1}^k \pi_l^{(t)} p(x_i|\hat{\theta}_l^{(t)})$  and

$$\nu_{ij} = \frac{\pi_j^{(t)} p(x_i|\hat{\theta}_j^{(t)})}{\sum_{m=1}^k \pi_m^{(t)} p(x_i|\hat{\theta}_m^{(t)})} \quad (7)$$

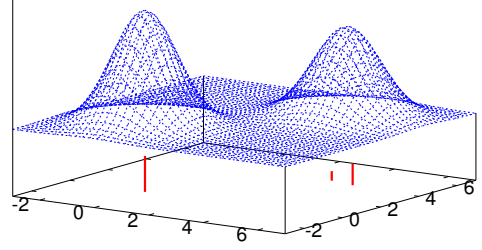


Figure 1. Example of the estimated mixture for  $\beta = -0.2$  and  $\sigma^2 = 1$ . Corresponding mixing distributions are illustrated in the x-y planes where the location and the height of the red lines are respectively the mean parameter  $\hat{\theta}_l$  and the weight  $\hat{\pi}_l$  of each component.

is the posterior probability that the data point  $x_i$  is assigned to the cluster center  $\hat{\theta}_l$ .

We can prove for  $\beta \leq 0$  that the above update monotonically decreases the objective  $F_\beta$  since this minimization is expressed by the double minimization over  $\{\pi_l, \hat{\theta}_l\}_{l=1}^k$  and  $\{p_i\}_{i=1}^n$  from eq.(3). However, the similar proof does not apply for  $\beta > 0$ . Hence, we switch to another update rule for  $\beta > 0$ , which is omitted in this paper.

### 4. EXPERIMENTS

In this section, we demonstrate the properties of the estimation method by a numerical simulation focusing on the case of 2-dimensional Gaussian mixtures where

$$p(x|\theta) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x - \theta\|^2}{2\sigma^2}\right). \quad (8)$$

We generated synthetic data by the true distribution,

$$p^*(x) = \frac{1}{2} N(x|\theta_1^*, I_2) + \frac{1}{2} N(x|\theta_2^*, I_2), \quad (9)$$

where  $\theta_1^* = (0, 0)^T$ ,  $\theta_2^* = (4, 4)^T$  and  $N(x|\theta, \sigma^2 I_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x-\theta\|^2}{2\sigma^2}\right)$  is the Gaussian density function.

We assumed that the kernel width  $\sigma^2$  in eq.(8) was known and  $p(x|\theta)$  was set to  $N(x|\theta, I_2)$ . Let  $\hat{q}(\theta)$  be an estimated mixing distribution. The optimal mixing distribution  $q(\theta)$  is given by  $\frac{1}{2}\delta(\theta-\theta_1^*) + \frac{1}{2}\delta(\theta-\theta_2^*)$  in this case. An example of the estimated mixture model for  $\beta = -0.2$  and  $\sigma^2 = 1$  is demonstrated in Figure 1.

Figure 2(a) and Figure 2(b) respectively show the training error,  $\frac{1}{n} \sum_{i=1}^n \log \frac{p^*(x_i)}{\int p(x_i|\theta)\hat{q}(\theta)d\theta}$ , and the generalization error,  $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \log \frac{p^*(\tilde{x}_i)}{\int p(\tilde{x}_i|\theta)\hat{q}(\theta)d\theta}$ , for test data  $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$  generated from the true distribution (9). All results were averaged over 100 trials for different data sets generated by (9). The number of training data is  $n = 50$  and that of test data is  $\tilde{n} = 200000$ . We also applied the original version of the algorithm in [2], where only  $\{\pi_l\}$  are updated by the EM algorithm with the weight  $p_i$  in eq.(4) for each sample in Step 4. These results are indicated as ‘‘means fixed’’. We see that the average training error takes the minimum at  $\beta = 0$  as expected while the average generalization error is minimized around  $\beta = -0.2$ .



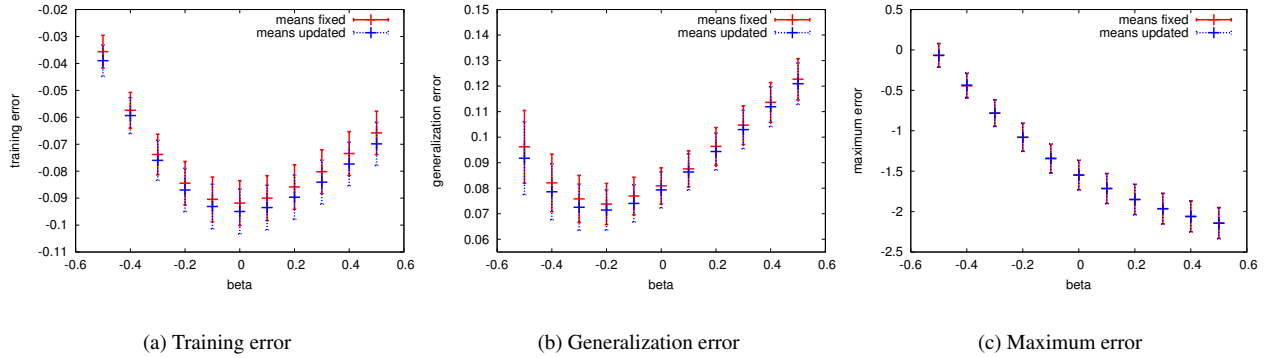


Figure 2. Training error (a), generalization error (b) and maximum error (c) against  $\beta$ . The error bars show 95% confidence intervals.

Figure 2(c) shows the average of the maximum error,  $\max_i (-\log \int p(x_i|\theta)\hat{q}(\theta)d\theta) - \max_i (-\log p^*(x_i))$ , which corresponds to the objective function of the KVQ. As expected, the monotone decrease of it with respect to  $\beta$  implies the estimation approaches the KVQ as  $\beta \rightarrow \infty$ .

In Figure 3, we show the number of estimated components remaining after the elimination of components with sufficiently small mixing proportions (less than  $\frac{1}{n^2}$ ). Since it strongly depends on  $\epsilon$ , we also applied hard assignments to cluster centers for each data point and counted the number of hard clusters, which is also plotted in Figure 3. Here, each point  $x_i$  is assigned to the cluster center  $\hat{\theta}_l$  that maximizes the posterior probability (7). The number of

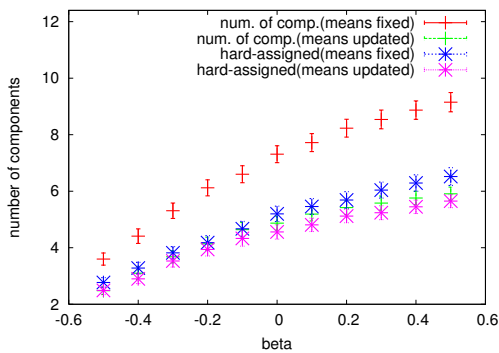


Figure 3. Number of components (cross) and number of hard clusters (asterisk) against  $\beta$ .

components  $\hat{k}$  as well as that of hard clusters increase as  $\beta$  becomes larger. This reduces the average generalization error when  $\beta$  takes slightly negative value as we just observed in Figure 2(b).

## 5. CONNECTION TO RATE-DISTORTION PROBLEM

The rate-distortion (RD) problem encoding the source random variable  $X$  with density  $p^*(x)$  to the output  $\Theta$  is reformulated to solving the following optimization problem

[4, 5],

$$\inf_q - \int p^*(x) \log \int q(\theta) \exp(sd(x, \theta)) d\theta dx. \quad (10)$$

Here  $d(x, \theta)$  is the distortion measure and  $s$  is a Lagrange multiplier. It provides the slope of a tangent to the RD curve and hence has one-to-one correspondence with a point on the RD curve. This problem reduces to the MLE ( $F_\beta(q)$  when  $\beta = 0$ ) with  $p(x|\theta) \propto \exp(sd(x, \theta))$  if the source  $p^*(x)$  is replaced with the empirical distribution. In the case of the Gaussian mixture with  $d(x, \theta) = \|x - \theta\|^2$ ,  $s$  specifies the kernel width by  $\sigma^2 = -\frac{1}{2s}$ .

For general  $\beta$ , the expression (3) and the optimal output distribution  $\hat{q}(\theta) = \sum_{l=1}^{\hat{k}} \hat{\pi}_l \delta(\theta - \hat{\theta}_l)$  imply the RD function of the source,  $\sum_{i=1}^n p_i \delta(x - x_i)$ , with the rate

$$\sum_{i=1}^n \sum_{l=1}^{\hat{k}} p_i \nu_{il} \log \frac{\nu_{il}}{\sum_{j=1}^n p_j \nu_{jl}},$$

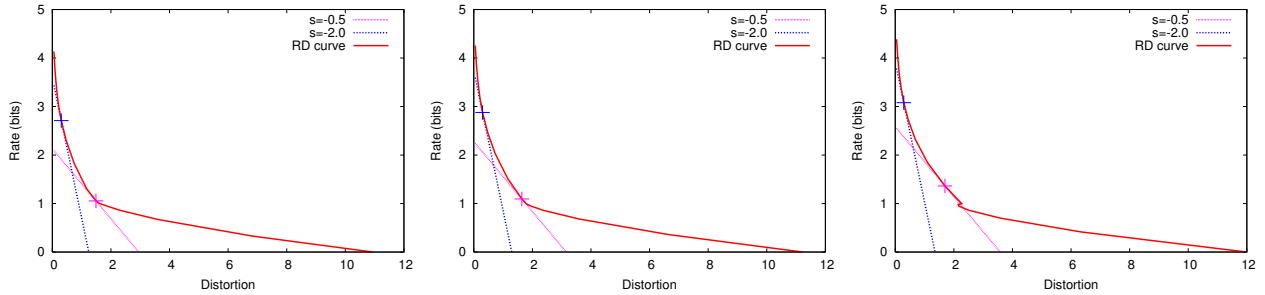
and the average distortion

$$\sum_{i=1}^n \sum_{l=1}^{\hat{k}} p_i \nu_{il} d(x_i, \hat{\theta}_l),$$

where  $\nu_{il}$  is the posterior probability defined by eq.(7). Since the rate is the mutual information between  $X$  and  $\Theta$ , it is bounded from above by the entropy,  $-\sum_{l=1}^{\hat{k}} \hat{\pi}_l \log \hat{\pi}_l$  and further by  $\log \hat{k}$ . However, the source depends on  $p_i$ , which depends on  $q(\theta)$  as in eq.(4) and hence the above pair of rate and distortion does not necessarily inherit properties of the usual RD function such as convexity.

Figure 4 demonstrates examples of RD functions obtained by the minimization of  $F_\beta(q)$  for  $\beta = -0.2$ ,  $\beta = 0$  and  $\beta = 0.5$  in the case of the Gaussian mixture used in Section 4.

The three curves show similar behavior such as a monotone decreasing trend although only that for  $\beta = 0.5$  loses convexity. This suggests the usage of the RD curve for determining the kernel width  $\sigma^2$ , e.g., by prespecifying a desired rate or average distortion. If we keep the desired rate or distortion to determine  $\sigma^2$  for different choices of  $\beta$ , then  $\beta$  can be chosen among them for example by CV.



(a) Rate-distortion curve for  $\beta = -0.2$ .

(b) Rate-distortion curve for  $\beta = 0.0$ .

(c) Rate-distortion curve for  $\beta = 0.5$ .

Figure 4. Examples of rate-distortion curves. The lines with slope  $s$  passing through the point corresponding to  $s$  (cross) are also illustrated for  $s = -0.5$  (magenta) and  $s = -2.0$  (blue). The rate is scaled by  $\log 2$  to yield bits.

## 6. EXTENSION TO OTHER CONVEX OBJECTIVE FUNCTIONS

The proposed algorithm in Section 3.2 is based on the decoupled approach developed in [2]. The general objective function considered in [2] includes the MLE and the KVQ to estimate  $q(\theta)$ . We proved in Section 3.1 by extending Lindsay’s theorem that the estimated  $q(\theta)$  is a discrete distribution consisting of distinct support points no more than  $n$ , the number of training data. This statement can be generalized to other objective functions as long as they are convex with respect to  $\mathbf{r} = (r_1, \dots, r_n)$  and hence to  $q(\theta)$ . More specifically, the following four objective functions are demonstrated as examples in [2]. Here,  $\rho = \min_i r_i$  and  $C$  is a constant.

1. MLE:  $-\sum_{i=1}^n \log r_i$
2. KVQ:  $-\rho$
3. Margin-minus-variance:  
 $-\rho + \frac{C}{n} \sum_{i=1}^n (r_i - \rho)^2$
4. Mean-minus-variance:  
 $-\frac{1}{n} \sum_{i=1}^n r_i + \frac{C}{n} \sum_{i=1}^n \left( r_i - \frac{1}{n} \sum_{j=1}^n r_j \right)^2$

The objective function  $F_\beta$  in eq.(2) combines the first two objectives by the parameter  $\beta$ . The other two objectives above are convex with respect to  $\mathbf{r}$  as well and hence can be proven to have optimal discrete distributions  $q(\theta)$  with support size no more than  $n$ . Note that since  $\mathbf{r}$  is a linear transformation of  $q(\theta)$ , the convexity on  $\mathbf{r}$  is equivalent to that on  $q(\theta)$  as long as  $q(\theta)$  appears in the objective function only with the form of  $r_i = \int p(x_i|\theta)q(\theta)d\theta$ . Furthermore, we have developed a simple algorithm for finite mixture models to minimize  $F_\beta$  in Section 3.3. Note that, to apply the general framework of Section 3.2 to specific objective functions, we need learning algorithms for optimizing them for finite mixture models.

## 7. CONCLUSION

We proposed an objective function for learning of mixture models, which unifies the MLE and the KVQ with the

parameter  $\beta$ . We proved that the optimal mixing distribution is a discrete distribution with distinct support points no more than the sample size and provided a simple algorithm to calculate it. We discussed the nature of the objective function in relation to the rate-distortion theory and demonstrated its less proneness to overfitting with an appropriate choice of the parameter.

## 8. REFERENCES

- [1] B. G. Lindsay, *Mixture Models: Theory, Geometry and Applications*, Institute of Mathematical Statistics, 1995.
- [2] S. Nowozin and G. Bakir, “A decoupled approach to exemplar-based unsupervised learning,” in *Proceedings of the 24th International Conference on Machine Learning (ICML 2008)*, 2008.
- [3] M. Tipping and B. Scholkopf, “A kernel approach for vector quantization with guaranteed distortion bounds,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2001.
- [4] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [5] D. Lashkari and P. Golland, “Convex clustering with exemplar-based models,” in *Advances in Neural Information Processing Systems 19*, 2007.



# EFFICIENT MESSAGE-PASSING FOR DISTRIBUTED QUADRATIC OPTIMIZATION

Guoqiang Zhang and Richard Heusdens

Department of Intelligent Systems  
 Delft University of Technology  
 Delft, the Netherlands  
 {g.zhang-1,r.heusdens}@tudelft.nl

## ABSTRACT

Distributed quadratic optimization (DQO) has found many applications in computer science and engineering. In designing a message-passing algorithm for DQO, the basic idea is to decompose the quadratic function into a set of local functions with respect to a graphic model. The nodes in the graph send local information of the quadratic function in message-form to their neighbors iteratively until reaching the global optimal solution. The efficiency of a message-passing algorithm depends on its computational complexity, the number of parameters to be transmitted, and its convergence speed. In this work, we study several message-passing algorithms for comparison. In particular, we consider the Jacobi-relaxation algorithm, the generalized linear coordinate descent (GLiCD) algorithm and the min-sum-min algorithm.

## 1. INTRODUCTION

In this work, we consider solving the quadratic optimization problem in a distributed fashion, namely

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \left( \frac{1}{2} x^\top J x - h^\top x \right), \quad (1)$$

where the quadratic matrix  $J$  is real symmetric positive definite and  $x$  is a real vector in  $n$ -dimensional space. It is known that the optimal solution is given by  $x^* = J^{-1}h$ . We suppose that the quadratic matrix  $J$  is sparse and the dimensionality  $n$  is large. In this situation, the direct computation (without using the sparse structure of  $J$ ) of the optimal solution may be expensive and unscalable. The research challenge is how to exploit the sparse geometry of  $J$  to efficiently obtain the optimal solution.

A common approach that exploits the sparsity of  $J$  is to associate the function  $f(x)$  with an undirected graph  $G = (V, E)$ . That is, the graph has a node for each variable  $x_i$  and an edge between node  $i$  and  $j$  only if the element  $J_{ij}$  is nonzero. By doing so, the sparsity of  $J$  is fully captured by the graph. As a consequence, the function can be decomposed with respect to  $G = (V, E)$  as

$$f(x) = \sum_{i \in V} f_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j), \quad (2)$$

where each edge-function  $f_{ij}(x_i, x_j)$  characterizes the interaction of  $x_i$  and  $x_j$  as specified by  $J_{ij}$ . With the graphic

model (2), distributed quadratic optimization (DQO) boils down to how to spread the global information of  $(J, h)$  in (1) over the graph efficiently by exchanging local information between neighboring nodes.

DQO over graphic models has found many applications in computer science and engineering in the past. Some applications are motivated by emerging parallel computational architectures (e.g., multicore CPUs and GPUs [1]), such as support vector machine [2] and channel coding [3, 4]. Other applications are motivated from the distributed nature carried by the problem, such as distributed speech enhancement in wireless microphone networks [5], distributed Kalman filter [6] and multiuser detection [7].

## 2. ALGORITHM COMPARISON

In the literature, the Jacobi algorithm is a classic method for solving the quadratic problem over the associated graph [8]. At each iteration, the algorithm performs node-oriented minimizations over all the nodes in the graph, of which the messages are in a form of linear functions (see Table 1). It is known that when the matrix  $J$  is walk-summable<sup>1</sup>, the Jacobi algorithm converges to the optimal solution [9, 10]. To fix the convergence for a general matrix  $J$ , the Jacobi algorithm was under-relaxed by incorporating an estimate of  $x^*$  from last iteration in computing a new estimate (see Table 1). It is well known that the Jacobi-relaxation algorithm possesses a guaranteed convergence if the relaxation parameter is properly chosen [8]. For the above two algorithms, once a node-estimate is updated, this estimate is broadcast to all its neighbors. Because the information transmitted is general, and not edge-specific, the two algorithms are known to converge slowly [8].

To accelerate the convergence of the Jacobi algorithm, we proposed the linear coordinate descent (LiCD) algorithm [11]. At each iteration, the LiCD algorithm performs pairwise minimizations over all the edges in the graph, of which the messages are in a form of linear functions (see Table 1). As shown in [11], if the quadratic matrix  $J$  is walk-summable, the LiCD algorithm converges to the optimal solution. Inspired by the Jacobi-relaxation

<sup>1</sup>A positive definite matrix  $J \in \mathbb{R}^{n \times n}$ , with all ones on its diagonal, is walk-summable if the spectral radius of the matrix  $\bar{R}$ , where  $R = I - J$  and  $\bar{R} = [R_{ij}]_{i,j=1}^n$ , is less than one (i.e.,  $\rho(\bar{R}) < 1$ ). We note that if the matrix  $J$  is diagonally dominant, it is also walk-summable.

$J$ is walk-summable	$J$ is general
Jacobi Alg.: * node-oriented minimization * linear message	Jacobi-relaxation Alg.: * introduce feedback in Jacobi Alg.
LiCD Alg.: * pairwise minimization * linear message	GLiCD Alg.: * introduce feedback in LiCD Alg.
min-sum Alg.: * pairwise minimization * quadratic message	min-sum-min Alg.: * introduce feedback in min-sum Alg.

Table 1. Algorithm comparison.

algorithm, we also extended the LiCD algorithm by incorporating feedback from last iteration in computing new messages in [12]. We name the new algorithm as the *generalized LiCD* (GLiCD) algorithm. The GLiCD algorithm was shown in [12] to converge to the optimal solution for a general matrix  $J$  when the amount of feedback signal is set to be large enough. For both the LiCD and the GLiCD algorithms, each node computes and transmits edge-specific information instead of broadcasting some common parameters to all its neighbors. Such edge-specific operation helps to spread the global information of  $(J, h)$  over the graph more effectively.

An alternative scheme for solving the quadratic problem is by using the framework of probability theory [13]. The optimal solution  $x^*$  is viewed as the mean value of a random vector  $x \in \mathbb{R}^n$  with Gaussian distribution

$$p(x) \propto \exp\left(-\frac{1}{2}x^\top Jx + h^\top x\right). \quad (3)$$

The min-sum algorithm is one popular approach to estimate both the mean value  $x^* = J^{-1}h$  and individual variances [14]. At each iteration, the algorithm essentially performs pairwise minimizations over all the edges in the graph, of which the messages are in a form of quadratic functions (see Table 1). For a graph with a tree-structure, the min-sum algorithm converges to the optimal solution in finite steps [14]. The question of convergence for loopy graphic models has been proven difficult. In [9, 10], it was shown when the matrix  $J$  is walk-summable, the min-sum algorithm converges to the optimal solution.

Due to the fact that the min-sum algorithm may fail a general matrix  $J$ , we proposed the min-sum-min algorithm [15] recently. The derivation of the min-sum-min algorithm follows the line of work in [12] for the GLiCD algorithm. Similarly to the GLiCD algorithm, the basic idea of the min-sum-min algorithm is to incorporate feedback from last iteration in computing new messages. We have shown in [15] that if the amount of the feedback is large enough, the min-sum-min algorithms converges to the optimal solution. We note that for the min-sum and the min-sum-min algorithms, each node computes and transmits edge-specific information to its neighbors, which is similar to that of the LiCD and the GLiCD algorithms.

The main properties of the above algorithms are summarized in Table 1. One observes that the Jacobi and the

LiCD algorithms share the property that their messages are in the form of linear functions. On the other hand, the LiCD and the min-sum algorithms share the property that both algorithms perform pairwise minimization at each iteration. From the viewpoint of minimization strategies and message-forms, the LiCD algorithm acts as an intermediate method between the Jacobi and the min-sum algorithms. As is analyzed in [11], the computational complexities of the three algorithms at each iteration are in the order of

$$\text{Jacobi Alg.} \rightarrow \text{LiCD Alg.} \rightarrow \text{min-sum Alg.}$$

where the min-sum algorithm is most expensive for implementation.

### 3. UNIFIED MESSAGE-PASSING FRAMEWORK

We note that all the algorithms listed in Table 1 share a unified message-passing framework despite the fact that different minimization strategies and message-forms are applied in the algorithms. We present the unified message-passing framework in the following.

Consider the quadratic optimization problem (1). We may assume, without loss of generality, that  $J$  is of unit-diagonal. The local node and edge functions for the graph  $G = (V, E)$  can be constructed as

$$f_i(x_i) = \frac{1}{2}x_i^2 - h_i x_i \quad i \in V \quad (4)$$

$$f_{ij}(x_i, x_j) = J_{ij}x_i x_j \quad (i, j) \in E. \quad (5)$$

An edge exists between node  $i$  and  $j$  in the graph only if  $J_{ij} \neq 0$ . As a consequence, a sparse matrix  $J$  leads to a sparse graph  $G = (V, E)$ . We use  $N(i)$  to denote the set of all neighbors of node  $i \in V$ . The set  $N(i) \setminus j$  excludes the node  $j$  from  $N(i)$ . For each edge  $(i, j) \in E$ , we use  $[j, i]$  and  $[i, j]$  to denote its two directed edges. Correspondingly, we denote the set of all directed edges of the graph as  $\vec{E}$ .

A message-passing algorithm exchanges information between neighboring nodes iteratively until reaching consensus. In particular, at time  $t$ , each node  $j$  collects a set of messages  $\{m_{v \rightarrow j}^{(t)}(x_j) | v \in N(j)\}$  and a set of estimates  $\{\hat{x}_{j|v}^{(t)}, v \in N(j)\}$  of  $x_j^*$  by cooperating with its neighbors. We note that for a directed edge  $[v, j] \in \vec{E}$ , the

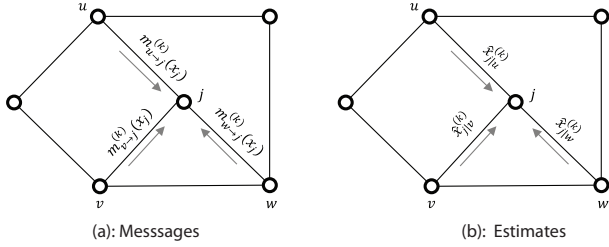


Figure 1. An example of the information-flow for node  $j$  at time step  $k$ .

associated message  $m_{v \rightarrow j}^{(t)}(x_i)$  and estimate  $\hat{x}_{j|v}$  are obtained by combining the local information of node  $v$  and  $j$  at time  $t - 1$  (see Fig. 1). For the Jacobi and the Jacobi-relaxation algorithms, the elements in  $\{\hat{x}_{j|v}^{(t)}, v \in N(j)\}$  for each node  $j$  are identical, since both algorithms perform node-oriented minimizations.

Given the messages at time  $t$ , one can define new local functions as

$$f_i^{(t)}(x_i) = f_i(x_i) + \sum_{u \in N(i)} m_{u \rightarrow i}^{(t)}(x_i) \quad i \in V$$

$$f_{ij}^{(t)}(x_i, x_j) = \left[ f_{ij}(x_i, x_j) - m_{j \rightarrow i}^{(t)}(x_i) - m_{i \rightarrow j}^{(t)}(x_j) \right] \quad (i, j) \in E$$

By summing up all the new local functions, it is straightforward that

$$f(x) = \sum_{i \in V} f_i^{(t)}(x_i) + \sum_{(i,j) \in E} f_{ij}^{(t)}(x_i, x_j). \quad (6)$$

Thus, the overall objective function remains the same. The new local functions can be viewed as a reformulation of the objective function.

The key part of a message-passing algorithm is the derivation of the updating expressions for  $\{(m_{j \rightarrow i}^{(t+1)}(x_i), \hat{x}_{i|j}^{(t+1)}), [j, i] \in \vec{E}\}$  given the information at time  $t$ . Note that for each node  $i$ , the estimates  $\{\hat{x}_{i|u}^{(t)}, u \in N(i)\}$  provide information about the optimal solution  $x_i^*$ . Thus, the estimates can be used as feedback in computing new messages and estimates in next iteration if necessary. An iterative algorithm converges to the optimal solution  $x^*$  if

$$\lim_{t \rightarrow \infty} \hat{x}_{i|j}^{(t)} = x_i^*, [j, i] \in \vec{E}. \quad (7)$$

Different iterative algorithms can be derived by choosing different minimization strategies and message-forms (see Table 1). As an example, we briefly present the Jacobi-relaxation algorithm in the following for demonstration. At time  $t$ , each node  $i$  keeps track of an estimate  $\hat{x}_i^{(t)}$  of  $x_i^*$  and a set of linear messages  $\{m_{u \rightarrow i}^{(t)}(x_i) = J_{iu} \hat{x}_u^{(t)}\}$ . The estimate  $\hat{x}_i^{(t+1)}$  at time step  $t + 1$  is computed as [8]

$$\hat{x}_i^{(t+1)} = \min_{x_i} \left[ f_i^{(t)}(x_i) + \frac{\alpha}{2} (x_i - \hat{x}_i^{(t)})^2 \right] \quad i \in V, (8)$$

where the parameter  $\alpha \in \mathbb{R}$  controls the amount of feedback in computing  $\hat{x}_i^{(t+1)}$ . Note that the feedback in (8) is represented by a quadratic penalty function in terms of  $\hat{x}_i^{(t)}$ , which can be easily merged into the local function  $f_i^{(t)}(x_i)$ . By letting  $\alpha = 1 - \frac{1}{s}$ , the above expression can be reformulated as

$$\hat{x}_i^{(t+1)} = \min_{x_i} \left[ s f_i(x_i) + \sum_{u \in N(i)} s J_{ui} \hat{x}_u^{(t)} + \frac{1-s}{2} (x_i - \hat{x}_i^{(t)})^2 \right] \quad i \in V.$$

In the literature,  $s$  is named as the relaxation parameter. When  $s = 1$  (or equivalently,  $\alpha = 0$ ), the Jacobi-relaxation algorithm reduces to the Jacobi algorithm. For a general matrix  $J$  in (1), the Jacobi-relaxation algorithm converges to the optimal solution  $x^*$  if the relaxation parameter  $s$  is sufficiently close to zero from above.

For those who are interested in the GLiCD and the min-sum-min algorithms, we refer the readers to [12] and [15]. Similarly to that of the Jacobi-relaxation, the feedbacks in the GLiCD and the min-sum-min algorithms are also represented by some quadratic penalty functions. The amount of feedback signal in the GLiCD algorithm or the min-sum-min algorithm is again controlled by a relaxation parameter.

#### 4. FUTURE WORK

We note that the Jacobi and the Jacobi-relaxation algorithms have a wide range of applications in practice. Naturally, it is worth trying other algorithms as listed in Table 1 for solving the same kind of problems. In future work, we will consider applying the GLiCD algorithm the min-sum-min algorithms for some practical problems.

#### 5. REFERENCES

- [1] Y. El-Kurdi, W. J. Gross, and D. Giannacopoulos, "Efficient implementation of gaussian belief propagation solver for large sparse diagonally dominant linear systems," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 471–474, 2012.
- [2] D. Bickson, D. Dolev, and E. Yom-Tov, "A Gaussian belief propagation solver for large scale Support Vector Machines," in *5th European Conference on Complex Systems*, Sept. 2008.
- [3] H. Uchikawa, B. M. Kurkoski, K. Kasai, and K. Sakaniwa, "Iterative Encoding with Gauss-Seidel Method for Spatially-Coupled Low-Density Lattice Codes," in *Proc. IEEE Int. Symp. Information Theory*, MIT Campus, USA, 2012.
- [4] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. Information Theory*, vol. 54, pp. 1561–1585, Apr. 2008.
- [5] R. Heusdens, G. Zhang, R. C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR Beamforming for (Wireless) Microphone Networks Using

Message Passing,” accepted by *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2012.

- [6] D. Bickson, O. Shental, and D. Dolev, “Distributed Kalman Filter via Gaussian Belief Propagation,” in *the 46th Allerton Conf. on Communications, Control and Computing*, 2008.
- [7] D. Bickson, O. Shental, P. H. Siegel, J. K. Wolf, and D. Dolev, “DGaussian belief propagation based multiuser detection,” in *In IEEE Int. Symp. on Inform. Theory (ISIT)*, July 2008, pp. 1878–1882.
- [8] D. P. Bertsekas and J. N. Tsitsikis, *Parallel and distributed Computation: Numerical Methods*, Belmont, MA: Athena Scientific, 1997.
- [9] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, “Walk-sum Interpretation and Analysis of Gaussian Belief Propagation,” in *Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, 2006, vol. 18.
- [10] D. M. Malioutov, J. K. Johnson, and A. S. Will-sky, “Walk-Sums and Belief Propagation in Gaussian Graphical Models,” *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.
- [11] G. Zhang and R. Heusdens, “Linear Coordinate-Descent Message-Passing for Quadratic Optimization,” appearing in *Neural Computation*.
- [12] G. Zhang and R. Heusdens, “Convergence of Generalized Linear Coordinate-Descent Message-Passing for Quadratic Optimization,” in *Proc. IEEE International Symposium on Information Theory*, June 2012.
- [13] S.L. Lauritzen, *Graphical Models*, Oxford University Press, 1996.
- [14] J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,” *Morgan Kaufman Publishers*, 1988.
- [15] G. Zhang and R. Heusdens, “Convergence of Min-Sum-Min Message-Passing for Quadratic Optimization,” in preparation for submission.

# GENERALISED ENTROPIES AND ASYMPTOTIC COMPLEXITIES OF LANGUAGES

*Yuri Kalnishkan, Michael V. Vyugin, and Vladimir Vovk*

Computer Learning Research Centre and Department of Computer Science,  
Royal Holloway, University of London,  
Egham, Surrey, TW20 0EX, United Kingdom

## ABSTRACT

The talk explores connections between asymptotic complexity and generalised entropy. Asymptotic complexity of a language (a language is a set of finite or infinite strings) is a way of formalising the complexity of predicting the next element in a sequence: it is the loss per element of a strategy asymptotically optimal for that language. Generalised entropy extends Shannon entropy to arbitrary loss functions; it is the optimal expected loss given a distribution on possible outcomes. It turns out that the set of tuples of asymptotic complexities of a language w.r.t. different loss functions can be described by means of generalised entropies corresponding to the loss functions.

## 1. INTRODUCTION

The complete version of this paper has been accepted to *Information and Computation*. An earlier version [1] appeared in conference proceedings.

We consider the following on-line learning scenario: given a sequence of previous outcomes  $x_1, x_2, \dots, x_{n-1}$ , a prediction strategy is required to output a prediction  $\gamma_n$  for the next outcome  $x_n$ .

We assume that outcomes belong to a finite *outcome space*  $\Omega$ . Predictions may be drawn from a compact *prediction space*  $\Gamma$ . A loss function  $\lambda : \Omega \times \Gamma \rightarrow [0, +\infty]$  is used to measure the discrepancy between predictions and actual outcomes; it is assumed to be continuous. The triple  $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$  describing the prediction environment is called a game.

The performance of a strategy  $\mathfrak{S}$  on a finite string  $\mathbf{x} = (x_1 x_2 \dots x_n)$  is measured by the cumulative loss  $\text{Loss}_{\mathfrak{S}}(\mathbf{x}) = \sum_{i=1}^n \lambda(x_i, \gamma_i)$ . Different aspects of this prediction framework have been extensively studied; see [2] for an overview.

One is tempted to define complexity of a string as the loss of an optimal strategy so that elements of “simple” strings  $\mathbf{x}$  are easy to predict and elements of “complicated” strings are hard to predict and large loss is incurred. However this intuitive idea is difficult to implement formally because it is hard to define an optimal strategy. If  $\mathbf{x}$  is fixed, the strategy can be tailored to suffer the minimum possible loss on  $\mathbf{x}$  (0 for natural loss functions such as square, absolute, or logarithmic). If there is complete flexibility in the choice of  $\mathbf{x}$ , i.e., “anything can happen”, then every strategy can be tricked into suffering large loss

and being greatly outperformed by some other strategy on some sequences  $\mathbf{x}$ .

One approach to this problem is predictive complexity introduced in [3] and studied in [4, 5, 6]. This approach replaces strategies by the class of semi-computable super-loss processes. Under certain restrictions on  $\Gamma$  and  $\lambda$  this class has a natural optimal element. Predictive complexity of a finite string is defined up to a constant and is similar in many respects to Kolmogorov complexity; predictive complexity w.r.t. the logarithmic loss function equals the negative logarithm of Levin’s a priori semi-measure.

This paper takes a different approach and introduces asymptotic complexity, which is in some respects easier and more intuitive. It is defined for languages (infinite sets of finite strings and sets of infinite sequences) and it equals the asymptotically optimal loss per element. This idea leads to several versions of complexity that behave slightly differently. An important advantage of this approach is that asymptotic complexity exists for all loss functions  $\lambda$  thus eliminating the question of existence, still partly unsolved for predictive complexity. One can consider effective and polynomial-time versions of asymptotic complexity by restricting oneself to computable or polynomial-time computable strategies. The existence of corresponding asymptotic complexities follows trivially.

In this paper we study the following question. Let  $\mathfrak{G}_k = \langle \Omega, \Gamma_k, \lambda_k \rangle$ ,  $k = 1, 2, \dots, K$ , be games with the same finite set of outcomes  $\Omega$ . How do asymptotic complexities of a same set of finite or infinite sequences of elements of  $\Omega$  compare? We answer this question by describing the set

$$(\text{AC}_1(L), \text{AC}_2(L), \dots, \text{AC}_K(L)) \subseteq \mathbb{R}^K,$$

where  $\text{AC}_k$  is an asymptotic complexity w.r.t.  $\mathfrak{G}_k$  and  $L$  ranges over all non-trivial languages. The set turns out to have a simple geometric description in terms of the generalised entropy studied in [7]. The set depends on the type of asymptotic complexity and may be different for different complexities<sup>1</sup>.

For the Shannon entropy there are many results connecting it with complexity and Hausdorff dimension; see,

<sup>1</sup>Note that the statement of the main theorem in the conference version [1] of this paper was inaccurate in this respect. A corrected journal version will appear soon

e.g., Theorem 2.8.1 in [8] and [9]. This paper directly generalises the main result of [10].

The set depends on the type of asymptotic complexity and may be different for different complexities<sup>2</sup>.

## 2. ASYMPTOTIC COMPLEXITY

### 2.1. Finite Sequences

Let  $L \subseteq \Omega^*$  be a set of finite strings. We call the values

$$\overline{\text{AC}}(L) = \inf_{\mathfrak{A}} \limsup_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n}, \quad (1)$$

$$\underline{\text{AC}}(L) = \inf_{\mathfrak{A}} \liminf_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n} \quad (2)$$

the *upper* and *lower asymptotic complexity* of  $L$  w.r.t. the game  $\mathfrak{G}$ . We use subscripts for AC to specify a particular game if it is not clear from the context.

In this paper we are concerned only with infinite sets of finite sequences and asymptotic complexity of a finite or an empty language  $L \subseteq \Omega^*$  is undefined. Thus by assumption there are strings of infinitely many lengths in  $L$ .

Still there may be no strings of a certain length in  $L$ . Let us assume that the limits in (1) and (2) are taken over the subsequence  $n_1 < n_2 < \dots$  of values such that  $L \cap \Omega^{n_i} \neq \emptyset$ .

### 2.2. Infinite Sequences

There are two natural ways to define complexities of non-empty languages  $L \subseteq \Omega^\infty$ .

First we can extend the notions we have just defined. Indeed, for a nonempty set of infinite sequences consider the set of all finite prefixes of all its sequences. The language thus obtained is infinite and has upper and lower complexities. For the resulting complexities we shall retain the notation  $\overline{\text{AC}}(L)$  and  $\underline{\text{AC}}(L)$ . We refer to these complexities as *uniform*.

The second way is the following. Let

$$\overline{\overline{\text{AC}}}(L) = \inf_{\mathfrak{A}} \sup_{\mathbf{x} \in L} \limsup_{n \rightarrow +\infty} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x}|_n)}{n},$$

$$\underline{\underline{\text{AC}}}(L) = \inf_{\mathfrak{A}} \sup_{\mathbf{x} \in L} \liminf_{n \rightarrow +\infty} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x}|_n)}{n}.$$

We refer to this complexity as *non-uniform*.

The concept of asymptotic complexity generalises certain complexity measures studied in the literature. The concepts of predictability and dimension studied in [10] can be easily reduced to asymptotic complexity: the dimension is the lower non-uniform complexity w.r.t. a multidimensional generalisation of the logarithmic game and predictability equals  $1 - \underline{\text{AC}}$ , where  $\underline{\text{AC}}$  is the lower non-uniform complexity w.r.t. a multidimensional generalisation of the absolute-loss game.

<sup>2</sup>Note that the statement of the main theorem in the conference version of this paper was inaccurate in this respect. A corrected journal version will appear soon

## 3. OTHER DEFINITIONS

### 3.1. Entropy

Let  $\mathbb{P}(\Omega)$  be the set of probability distributions on  $\Omega$  of size  $M$ . The set  $\Omega$  is finite and we can identify  $\mathbb{P}(\Omega)$  with the standard  $(M - 1)$ -simplex

$$\mathbb{P}_M = \left\{ \left( p^{(0)}, p^{(1)}, \dots, p^{(M-1)} \right) \in [0, 1]^M \mid \sum_{i=0}^{M-1} p^{(i)} = 1 \right\}.$$

*Generalised entropy*  $H : \mathbb{P}(\Omega) \rightarrow \mathbb{R}$  is the infimum of expected loss over  $\gamma \in \Gamma$ , i.e., for

$$p^* = \left( p^{(0)}, p^{(1)}, \dots, p^{(M-1)} \right) \in \mathbb{P}(\Omega)$$

we have

$$H(p^*) = \min_{\gamma \in \Gamma} \mathbf{E}_{p^*} \lambda(\omega, \gamma) = \min_{\gamma \in \Gamma} \sum_{i=0}^{M-1} p^{(i)} \lambda(\omega^{(i)}, \gamma).$$

Since  $p^{(i)}$  can be 0 and  $\lambda(\omega^{(i)}, \gamma)$  can be  $+\infty$ , we need to resolve an ambiguity. Let us assume that in this definition  $0 \times (+\infty) = 0$ .

### 3.2. Sublattices and Subsemilattices

A set  $\mathcal{M} \subseteq \mathbb{R}^K$  is a *sublattice* of  $\mathbb{R}^K$  if for every  $x, y \in \mathcal{M}$  it contains their coordinate-wise greatest lower bound  $\min(x, y)$  and least upper bound  $\max(x, y)$ . Clearly, a sublattice of  $\mathbb{R}^K$  contains the coordinate-wise maximum and minimum of any finite subset. Similarly, a set  $\mathcal{M} \subseteq \mathbb{R}^K$  is an *upper subsemilattice* if for every  $x, y \in \mathcal{M}$  it contains their smallest upper bound  $\max(x, y)$ ; a set  $\mathcal{M} \subseteq \mathbb{R}^K$  is a *lower subsemilattice* if for every  $x, y \in \mathcal{M}$  it contains their largest lower bound  $\min(x, y)$ . In this paper we mostly use upper subsemilattices and therefore sometimes omit the word “upper” in what follows.

A *sublattice closure* of a set  $\mathcal{M} \subseteq \mathbb{R}^K$  is the smallest sublattice containing  $\mathcal{M}$ . Respectively, an *upper subsemilattice closure* of a set  $\mathcal{M} \subseteq \mathbb{R}^K$  is the smallest upper semilattice containing  $\mathcal{M}$  and a *lower subsemilattice closure* of a set  $\mathcal{M} \subseteq \mathbb{R}^K$  is the smallest lower subsemilattice containing  $\mathcal{M}$ . The sub(semi)lattice closure of  $\mathcal{M}$  exists and it is the intersection of all sub(semi)lattices containing  $\mathcal{M}$ . The sublattice closure contains the subsemilattice closures because each sublattice is a subsemilattice.

Note that the definitions are coordinate-dependent.

### 3.3. Weak Mixability

The results of this paper are valid for the so called weakly mixable games defined in [11]. A game  $\mathfrak{G}$  is weakly mixable if for every two prediction strategies  $\mathfrak{S}_1$  and  $\mathfrak{S}_2$  there is a prediction strategy  $\mathfrak{S}$  such that

$$\text{Loss}_{\mathfrak{S}}(\mathbf{x}) \leq \min(\text{Loss}_{\mathfrak{S}_1}(\mathbf{x}), \text{Loss}_{\mathfrak{S}_2}(\mathbf{x})) + \alpha(|\mathbf{x}|) \quad (3)$$

for all finite strings  $\mathbf{x}$ , where  $|\mathbf{x}|$  is the length of  $\mathbf{x}$  and  $\alpha(n) = o(n)$  as  $n \rightarrow \infty$ . It is shown in [11] that weak

mixability is equivalent to the convexity of the set of superpredictions w.r.t.  $\mathfrak{G}$ . In particular, if  $\Gamma$  is convex and  $\lambda$  is convex in predictions, weak mixability holds.

### 3.4. Effective Versions of Complexities

One can restrict the range of possible strategies to computable or polynomial-time computable and obtain effective and polynomial-time versions of the asymptotic complexities.

The concept of a computable strategy requires clarification. We will give a definition along the lines of [12]; see also [13, Sections 7 and 9.4].

A *dyadic* rational number is a number of the form  $m/2^n$ , where  $m$  is an integer and  $n$  is a positive integer. We call a triple  $\langle b, \mathbf{x}, \mathbf{y} \rangle$ , where  $b \in \mathbb{B}$  is a bit and  $\mathbf{x}, \mathbf{y} \in \mathbb{B}^*$  are binary strings, a *representation of a dyadic number*  $d$  if  $\mathbf{x}$  is the binary representation of a nonnegative integer  $m > 0$ ,  $\mathbf{y}$  is the binary representation of a nonnegative integer  $n > 0$ , and  $b$  represents a sign  $s$  (assume that  $s = 1$  if  $b = 1$  and  $s = -1$  if  $b = 0$ ) so that  $d = sm/2^n$ .

For every  $x \in \mathbb{R}$  define a set  $\text{CF}_x$  of dyadic Cauchy sequences exponentially converging to  $x$ , i.e., functions  $\phi_x$  from non-negative integers to dyadic numbers such that  $|\phi_x(n) - x| \leq 2^{-n}$  for all  $n$ . Any element of  $\text{CF}_x$  can be thought of as a dyadic representation of  $x$ .

Let  $\Omega$  be a finite set. A function  $f : \Omega^* \rightarrow \mathbb{R}$  is computable if there is a Turing machine that given a finite string  $\mathbf{x} = x_1x_2 \dots x_m \in \Omega^*$  and non-negative integer precision  $n$  outputs a representation of a dyadic number  $d$  such that  $|f(\mathbf{x}) - d| \leq 2^{-n}$ . In other words, for every  $\mathbf{x} \in \Omega^*$  the machine calculates a function from  $\text{CF}_{f(\mathbf{x})}$ . If there is a polynomial  $p(\cdot, \cdot)$  such that the machine always finishes work in  $p(m, n)$ , we say that  $f$  is polynomial-time computable. A function  $f = (f_1, f_2, \dots, f_k) : \Omega^* \rightarrow \mathbb{R}^k$  is (polynomial-time) computable if all its components  $f_1, f_2, \dots, f_k$  are (polynomial-time) computable.

A function  $f : M \rightarrow \mathbb{R}$ , where  $M \subseteq \mathbb{R}$ , is computable if there is an oracle Turing machine that given a non-negative integer precision  $n$  (as a binary string) and an oracle evaluating some  $\phi_x \in \text{CF}_x$  outputs a representation of a dyadic number  $d$  such that  $|f(x) - d| \leq 2^{-n}$ . If there is a polynomial  $p(\cdot)$  such that the machine finishes work in  $p(n)$  for all  $x \in M$ , we say that  $f$  is polynomial-time computable. Intuitively a machine can at any moment request a dyadic approximation of  $x$  up to  $2^{-m}$  and get it in no time. Computable and polynomial-time computable functions on  $M \subseteq \mathbb{R}^k$  and  $M \times \Omega^*$  to  $\mathbb{R}$  and  $\mathbb{R}^m$  are defined in a similar fashion.

We call a game  $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$  (*polynomial-time*) *computable* if  $\Gamma \subseteq \mathbb{R}^k$  is a closure of its interior and the function  $e^{-\lambda(\omega, \gamma)}$  is (polynomial-time) computable. Note that we do not postulate computability of  $\lambda$  itself because if would have implied boundedness of  $\lambda$ . A (*polynomial-time*) *computable strategy* w.r.t.  $\mathfrak{G}$  is a (polynomial-time) computable function  $\Omega^* \rightarrow \Gamma$ .

### 3.5. Computability and Weak Mixability

A (polynomial-time) computable game  $\mathfrak{G}$  will be called (*polynomial-time*) *computable very weakly mixable* if for all (polynomial-time) computable strategies  $\mathfrak{G}_1$  and  $\mathfrak{G}_2$  and  $\varepsilon > 0$  there is a (polynomial-time) computable strategy  $\mathfrak{G}$  such that

$$\text{Loss}_{\mathfrak{G}}(\mathbf{x}) \leq \min(\text{Loss}_{\mathfrak{G}_1}(\mathbf{x}), \text{Loss}_{\mathfrak{G}_2}(\mathbf{x})) + \varepsilon|\mathbf{x}| + \alpha_\varepsilon(|\mathbf{x}|)$$

for all finite strings  $\mathbf{x}$ , where  $\alpha_\varepsilon(n) = o(n)$  as  $n \rightarrow \infty$ .

It is not easy to formulate a simple criterion of computable mixability. The following rather general condition is sufficient. If a game  $\mathfrak{G}$  is (polynomial-time) computable, the prediction space  $\Gamma$  is convex, and the loss function  $\lambda(\omega, \gamma)$  is convex in the second argument, then  $\mathfrak{G}$  is (polynomial-time) computable weakly mixable.

If we add the requirement of boundness of  $\lambda$ , we can achieve an effective version of (3), but this is not necessary for the purpose of this paper.

## 4. MAIN RESULT

Consider  $K \geq 1$  games  $\mathfrak{G}_1, \mathfrak{G}_2, \dots, \mathfrak{G}_K$  with the same finite set of outcomes  $\Omega$ . Let  $H_k$  be  $\mathfrak{G}_k$ -entropy for  $k = 1, 2, \dots, K$ . The  $\mathfrak{G}_1/\mathfrak{G}_2/\dots/\mathfrak{G}_K$ -entropy set is the set  $\{(H_1(p), H_2(p), \dots, H_K(p)) \mid p \in \mathbb{P}(\Omega)\} \subseteq \mathbb{R}^K$ . The convex hull of the  $\mathfrak{G}_1/\mathfrak{G}_2/\dots/\mathfrak{G}_K$ -entropy set is called the  $\mathfrak{G}_1/\mathfrak{G}_2/\dots/\mathfrak{G}_K$ -entropy hull.

**Theorem 1.** *If games  $\mathfrak{G}_1, \mathfrak{G}_2, \dots, \mathfrak{G}_K$  ( $K \geq 1$ ) have the same finite outcome space  $\Omega$  and are weakly mixable, then the sublattice closure of the  $\mathfrak{G}_1/\mathfrak{G}_2/\dots/\mathfrak{G}_K$ -entropy hull coincides with the following sets (here  $\text{AC}_k$  is asymptotic complexity w.r.t.  $\mathfrak{G}_k$ ,  $k = 1, 2, \dots, K$ ):*

$$\left\{ \left( \underline{\text{AC}}_1(L), \underline{\text{AC}}_2(L), \dots, \underline{\text{AC}}_K(L) \right) \mid L \subseteq \Omega^* \text{ and } L \text{ is infinite} \right\};$$

$$\left\{ \left( \underline{\text{AC}}_1(L), \underline{\text{AC}}_2(L), \dots, \underline{\text{AC}}_K(L) \right) \mid L \subseteq \Omega^\infty \text{ and } L \neq \emptyset \right\};$$

$$\left\{ \left( \underline{\underline{\text{AC}}}_1(L), \underline{\underline{\text{AC}}}_2(L), \dots, \underline{\underline{\text{AC}}}_K(L) \right) \mid L \subseteq \Omega^\infty \text{ and } L \neq \emptyset \right\};$$

*the upper subsemilattice closure of the  $\mathfrak{G}_1/\mathfrak{G}_2/\dots/\mathfrak{G}_K$ -entropy hull coincides with the following sets:*

$$\left\{ \left( \overline{\text{AC}}_1(L), \overline{\text{AC}}_2(L), \dots, \overline{\text{AC}}_K(L) \right) \mid L \subseteq \Omega^* \text{ and } L \text{ is infinite} \right\};$$

$$\left\{ \left( \overline{\text{AC}}_1(L), \overline{\text{AC}}_2(L), \dots, \overline{\text{AC}}_K(L) \right) \mid \right. \\ \left. L \subseteq \Omega^\infty \text{ and } L \neq \emptyset \right\} ;$$

$$\left\{ \left( \overline{\overline{\text{AC}}}_1(L), \overline{\overline{\text{AC}}}_2(L), \dots, \overline{\overline{\text{AC}}}_K(L) \right) \mid \right. \\ \left. L \subseteq \Omega^\infty \text{ and } L \neq \emptyset \right\} .$$

If the games  $\mathfrak{G}_1, \mathfrak{G}_2, \dots, \mathfrak{G}_K$  are (polynomial-time) computable very weakly mixable, the same holds for effective and polynomial-time complexities.

The conference version [1] of the paper incorrectly claimed that all the sets of complexity tuples coincide with the upper subsemilattice closure of the entropy hull. This is not true because upper subsemilattice closure of the entropy hull may be different from the sublattice closure.

## 5. RECALIBRATION LEMMA

The key element of the proof is the following lemma:

**Lemma 1.** *Let  $\mathfrak{A}_1, \mathfrak{A}_2, \dots, \mathfrak{A}_K$  be prediction strategies for weakly mixable games  $\mathfrak{G}_1, \mathfrak{G}_2, \dots, \mathfrak{G}_K$  with the same set of outcomes  $\Omega$  of size  $M$ . Then for every weakly mixable game  $\mathfrak{G}$  and  $\varepsilon > 0$  there is a prediction strategy  $\mathfrak{S}$  and a function  $f : \mathbb{N} \rightarrow \mathbb{R}$  such that  $f(n) = o(n)$  as  $n \rightarrow \infty$  and for every finite string  $\mathbf{x} \in \Omega^*$  there are distributions  $p_1, p_2, \dots, p_N \in \mathbb{P}_M$  and  $q = (q_1, q_2, \dots, q_N) \in \mathbb{P}_N$  such that*

1. *for all  $k = 1, 2, \dots, K$  if  $H_k$  is the generalised entropy w.r.t.  $\mathfrak{G}_k$  then*

$$\sum_{i=1}^N q_i H_k(p_i) \leq \frac{\text{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\mathbf{x})}{|\mathbf{x}|} + \varepsilon ;$$

2. *if  $H$  is the generalised entropy w.r.t.  $\mathfrak{G}$  then*

$$\text{Loss}_{\mathfrak{S}}^{\mathfrak{G}}(\mathbf{x}) \leq |\mathbf{x}| \left( \sum_{i=1}^N q_i H(p_i) + \varepsilon \right) + f(|\mathbf{x}|) .$$

The idea behind the lemma can be described informally as follows. Consider a predictor outputting, say, the likelihood of a rain. Suppose that by analysing its past performance we have found a pattern of the following kind. Whenever the predictor outputs the value of 70%, it actually rains in 90% of cases. We can thus improve the predictor by *recalibrating* it: if we see the prognosis of 70%, we replace it by 90%. Generally speaking, we may observe that whenever a predictor outputs a prediction  $\gamma_1$ , a more appropriate choice would be  $\gamma_2$ . By outputting  $\gamma_1$ , the predictor signals us about a specific state of the nature; however,  $\gamma_2$  is a better prediction for this state. The loss per element of the optimised strategy is close to the generalised entropy w.r.t. some distribution and this leads to the first part of the lemma.

The intuitive interpretation of the second part is as follows. Predictions of (discretised) strategies allow us to split a string to several (generally speaking, not contiguous) substrings. The strategies tell us nothing of the behaviour of outcomes within the substrings so we can assume that inside each substring the outcomes are i.i.d. (independent identically distributed) and construct a new strategy exploiting this. The loss per element of the new strategy will be a convex combination of entropies w.r.t. the distributions of outcomes from the substrings and the new strategy will perform better or nearly as well as the original strategies.

## 6. REFERENCES

- [1] V. Vovk, Y. Kalnishkan and M.V. Vyugin, "Generalised entropy and asymptotic complexities of languages," in *20th Annual Conference on Learning Theory, COLT 2007*, 2007, pp. 293–307, Springer.
- [2] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*, Cambridge University Press, 2006.
- [3] V. Vovk and C. J. H. C. Watkins, "Universal portfolio selection," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 1998, pp. 12–23, ACM Press.
- [4] Y. Kalnishkan, "General linear relations among different types of predictive complexity," *Theoretical Computer Science*, vol. 271, pp. 181–200, 2002.
- [5] Y. Kalnishkan, V. Vovk, and M. V. Vyugin, "Loss functions, complexities, and the Legendre transformation," *Theoretical Computer Science*, vol. 313, no. 2, pp. 195–207, 2004.
- [6] Y. Kalnishkan, V. Vovk, and M. V. Vyugin, "How many strings are easy to predict?," *Information and Computation*, vol. 201, no. 1, pp. 55–71, 2005.
- [7] P. D. Grünwald and A. P. Dawid, "Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory," *The Annals of Statistics*, vol. 32, no. 4, pp. 1367–1433, 2004.
- [8] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 3rd edition, 2008.
- [9] B. Ya. Ryabko, "Noiseless coding of combinatorial sources, hausdorff dimension, and Kolmogorov complexity," *Problems of Information Transmission*, vol. 22, no. 3, pp. 170–179, 1986.
- [10] L. Fortnow and J. H. Lutz, "Prediction and dimension," *Journal of Computer and System Sciences*, vol. 70, no. 4, pp. 570–589, 2005.
- [11] Y. Kalnishkan and M. V. Vyugin, "The weak aggregating algorithm and weak mixability," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1228–1244, 2008.
- [12] Ker-I Ko, *Complexity theory of real functions*, Birkhäuser, 1991.
- [13] K. Weihrauch, *Computable Analysis*, Springer, 2000.



# INFORMATIONAL AND COMPUTATIONAL EFFICIENCY OF SET PREDICTORS

Vladimir Vovk

Computer Learning Research Centre  
Department of Computer Science  
Royal Holloway, University of London  
United Kingdom

## ABSTRACT

There are two methods of set prediction that are provably valid under the assumption of randomness: transductive conformal prediction and inductive conformal prediction. The former method is informationally efficient but often lacks computational efficiency. The latter method is, vice versa, computationally efficient but less efficient informationally. This talk discusses a new method, which we call cross-conformal prediction, that combines informational efficiency of transductive conformal prediction with computational efficiency of inductive conformal prediction. The downside of the new method is that its validity is an empirical rather than mathematical fact.

## 1. INTRODUCTION

The method of (transductive) conformal prediction produces set predictions that are automatically valid in the sense that their unconditional coverage probability is equal to or exceeds a preset confidence level ([1], Chapter 2). A more computationally efficient method of this kind is that of inductive conformal prediction ([2], [1], Section 4.1, [3]). However, inductive conformal predictors are typically less informationally efficient, in the sense of producing larger prediction sets as compared with conformal predictors. Motivated by the method of cross-validation, this talk explores a hybrid method, which we call cross-conformal prediction.

We are mainly interested in the problems of classification and regression, in which we are given a training set consisting of examples, each example consisting of an object and a label, and asked to predict the label of a new test object; in the problem of classification labels are elements of a given finite set, and in the problem of regression labels are real numbers. If we are asked to predict labels for more than one test object, the same prediction procedure can be applied to each test object separately. In this introductory section and in our empirical studies we consider the problem of binary classification, in which labels can take only two values, which we will encode as 0 and 1.

---

The empirical studies described in this paper used the R system and the `glm` package written by Greg Ridgeway (based on the work of Freund, Schapire, and Friedman). This work was partially supported by the Cyprus Research Promotion Foundation.

We always assume that the examples (both the training examples and the test examples, consisting of given objects and unknown labels) are generated from an exchangeable probability measure (i.e., a probability measure that is invariant under permuting the examples). This *exchangeability assumption* is slightly weaker than the *assumption of randomness* that the examples are generated independently from the same probability measure.

The idea of conformal prediction is to try the two different labels, 0 and 1, for the test object, and for either postulated label to test the assumption of exchangeability by checking how well the test example conforms to the training set; the output of the procedure is the corresponding p-values  $p^0$  and  $p^1$ . Two standard ways to package the pair  $(p_0, p_1)$  are:

- Report the *confidence*  $1 - \min(p^0, p^1)$  and *credibility*  $\max(p^0, p^1)$ .
- For a given significance level  $\epsilon \in (0, 1)$  output the corresponding prediction set  $\{y \mid p^y > \epsilon\}$ .

In inductive conformal prediction the training set is split into two parts, the proper training set and the calibration set. The two p-values  $p^0$  and  $p^1$  are computed by checking how well the test example conforms to the calibration set. The way of checking conformity is based on a prediction rule found from the proper training set and produces, for each example in the calibration set and for the test example, the corresponding “conformity score”. The conformity score of the test example is then calibrated to the conformity scores of the calibration set to obtain the p-value. For details, see Section 2.

Inductive conformal predictors are usually much more computationally efficient than the corresponding conformal predictors. However, they are less informationally efficient: they use only the proper training set when developing the prediction rule and only the calibration set when calibrating the conformity score of the test example, whereas conformal predictors use the full training set for both purposes.

Cross-conformal prediction modifies inductive conformal prediction in order to use the full training set for calibration and significant parts of the training set (such as 80% or 90%) for developing prediction rules. The training set is split into  $K$  folds of equal (or almost equal) size.

For each  $k = 1, \dots, K$  we construct a separate inductive conformal predictor using the  $k$ th fold as the calibration set and the rest of the training set as the proper training set. Let  $(p_k^0, p_k^1)$  be the corresponding p-values. Next the two sets of p-values,  $p_k^0$  and  $p_k^1$ , are merged into combined p-values  $p^0$  and  $p^1$ , which are the result of the procedure.

In Section 3 we describe the method of cross-conformal prediction. Since we have no theoretical results about the validity of cross-conformal prediction in this talk, we rely on empirical studies involving the standard Spambase data set. Finally, we use the same data set to demonstrate the efficiency of cross-conformal predictors as compared with inductive conformal predictors. Section 4 states an open problem.

For the full version of this extended abstract, see [4].

## 2. INDUCTIVE CONFORMAL PREDICTORS

We fix two measurable spaces:  $\mathbf{X}$ , called the *object space*, and  $\mathbf{Y}$ , called the *label space*. The Cartesian product  $\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$  is the *example space*. A *training set* is a sequence  $(z_1, \dots, z_l) \in \mathbf{Z}^l$  of *examples*  $z_i = (x_i, y_i)$ , where  $x_i \in \mathbf{X}$  are the *objects* and  $y_i \in \mathbf{Y}$  are the *labels*. For  $S \subseteq \{1, \dots, l\}$ , we let  $z_S$  stand for the sequence  $(z_{s_1}, \dots, z_{s_n})$ , where  $s_1, \dots, s_n$  is the sequence of all elements of  $S$  listed in the increasing order (so that  $n := |S|$ ).

In the method of inductive conformal prediction, we split the training set into two non-empty parts, the *proper training set*  $z_T$  and the *calibration set*  $z_C$ , where  $(T, C)$  is a partition of  $\{1, \dots, l\}$ . An *inductive conformity measure* is a measurable function  $A : \mathbf{Z}^* \times \mathbf{Z} \rightarrow \mathbb{R}$  (we are interested in the case where  $A(\zeta, z)$  does not depend on the order of the elements of  $\zeta \in \mathbf{Z}^*$ ). The idea behind the *conformity score*  $A(z_T, z)$  is that it should measure how well the example  $z$  conforms to the proper training set  $z_T$ . A standard choice is

$$A(z_T, (x, y)) := \Delta(y, f(x)), \quad (1)$$

where  $f : \mathbf{X} \rightarrow \mathbf{Y}'$  is a prediction rule found from  $z_T$  as the training set and  $\Delta : \mathbf{Y} \times \mathbf{Y}' \rightarrow \mathbb{R}$  is a measure of similarity between a label and a prediction. Allowing  $\mathbf{Y}'$  to be different from  $\mathbf{Y}$  (usually  $\mathbf{Y}' \supset \mathbf{Y}$ ) may be useful when the underlying prediction method gives additional information to the predicted label; e.g., the MART procedure used in Section 3 gives the logit of the predicted probability that the label is 1.

The *inductive conformal predictor* (ICP) corresponding to  $A$  is defined as the set predictor

$$\Gamma^\epsilon(z_1, \dots, z_l, x) := \{y \mid p^y > \epsilon\}, \quad (2)$$

where  $\epsilon \in (0, 1)$  is the chosen *significance level* ( $1 - \epsilon$  is known as the *confidence level*), the *p-values*  $p^y$ ,  $y \in \mathbf{Y}$ , are defined by

$$p^y := \frac{|\{i \in C \mid \alpha_i \leq \alpha^y\}| + 1}{|C| + 1},$$

and

$$\alpha_i := A(z_T, z_i), \quad i \in C, \quad \alpha^y := A(z_T, (x, y)) \quad (3)$$

are the conformity scores. Given the training set and a test object  $x$  the ICP predicts its label  $y$ ; it *makes an error* if  $y \notin \Gamma^\epsilon(z_1, \dots, z_l, x)$ .

The random variables whose realizations are  $x_i, y_i, z_i, x, y, z$  will be denoted by the corresponding upper case letters  $(X_i, Y_i, Z_i, X, Y, Z)$ , respectively). The following proposition of validity is almost obvious.

**Proposition 1** ([1], Proposition 4.1). *If random examples  $Z_1, \dots, Z_l, Z = (X, Y)$  are exchangeable (i.e., their distribution is invariant under permutations), the probability of error  $Y \notin \Gamma^\epsilon(Z_1, \dots, Z_l, X)$  does not exceed  $\epsilon$  for any  $\epsilon$  and any inductive conformal predictor  $\Gamma$ .*

We call the property of inductive conformal predictors asserted in Proposition 1 unconditional validity since it is about the unconditional probability of error. Various conditional properties of validity are discussed in [5] and, in more detail, [6].

The family of prediction sets  $\Gamma^\epsilon(z_1, \dots, z_l, x)$ ,  $\epsilon \in (0, 1)$ , is just one possible way of packaging the p-values  $p^y$ . Another way, already discussed in Section 1 in the context of binary classification, is as the *confidence*  $1 - p$ , where  $p$  is the second largest p-value among  $p^y$ , and the *credibility*  $\max_y p^y$ . In the case of binary classification confidence and credibility carry the same information as the full set  $\{p^y \mid y \in \mathbf{Y}\}$  of p-values, but this is not true in general.

In our experiments reported in the next section we split the training set into the proper training set and the calibration set in proportion 2 : 1. This is the most standard proportion (cf. [7], p. 222, where the validation set plays a similar role to our calibration set), but the ideal proportion depends on the learning curve for the given problem of prediction (cf. [7], Figure 7.8). Too small a calibration set leads to a high variance of confidence (since calibrating conformity scores becomes unreliable) and too small a proper training set leads to a downward bias in confidence (conformity scores based on a small proper training set cannot produce confident predictions). In the next section we will see that using cross-conformal predictors improves both bias and variance (cf. Table 1).

## 3. CROSS-CONFORMAL PREDICTORS

*Cross-conformal predictors* (CCP) are defined as follows. The training set is split into  $K$  non-empty subsets (*folds*)  $z_{S_k}$ ,  $k = 1, \dots, K$ , where  $K \in \{2, 3, \dots\}$  is a parameter of the algorithm and  $(S_1, \dots, S_K)$  is a partition of  $\{1, \dots, l\}$ . For each  $k \in \{1, \dots, K\}$  and each potential label  $y \in \mathbf{Y}$  of the test object  $x$  find the conformity scores of the examples in  $z_{S_k}$  and of  $(x, y)$  by

$$\alpha_{i,k} := A(z_{S_k}, z_i), \quad i \in S_k, \quad \alpha_k^y := A(z_{S_k}, (x, y)), \quad (4)$$

where  $S_{-k} := \cup_{j \neq k} S_j$  and  $A$  is a given inductive conformity measure. The corresponding p-values are defined by

$$p^y := \frac{\sum_{k=1}^K |\{i \in S_k \mid \alpha_{i,k} \leq \alpha_k^y\}| + 1}{l + 1}. \quad (5)$$

Confidence and credibility are now defined as before; the set predictor  $\Gamma^\epsilon$  is also defined as before, by (2), where  $\epsilon > 0$  is another parameter.

The definition of CCPs parallels that of ICPs, except that now we use the whole training set for calibration. The conformity scores (4) are computed as in (3) but using the union of all the folds except for the current one as the proper training set. Calibration (5) is done by combining the ranks of the test example  $(x, y)$  with a postulated label in all the folds.

If we define the separate p-value

$$p_k^y := \frac{|\{i \in S_k \mid \alpha_{i,k} \leq \alpha_k^y\}| + 1}{|S_k| + 1}$$

for each fold, we can see that  $p^y$  is essentially the average of  $p_k^y$ . In particular, if each fold has the same size,  $|S_1| = \dots = |S_K|$ , a simple calculation gives

$$p^y = \bar{p}^y + \frac{K-1}{l+1} (\bar{p}^y - 1) \approx \bar{p}^y,$$

where  $\bar{p}^y := \frac{1}{K} \sum_{k=1}^K p_k^y$  is the arithmetic mean of  $p_k^y$  and the  $\approx$  assumes  $K \ll l$ .

We give calibration plots for 5-fold and 10-fold cross-conformal prediction taking  $K \in \{5, 10\}$  following the advice in [7] (who refer to Breiman and Spector’s and Kohavi’s work). In our experiments we use the popular Spambase data set. The size of the data set is 4601, and there are two labels: `spam`, encoded as 1, and `email`, encoded as 0.

We consider the conformity measure (1) where  $f$  is output by MART ([7], Chapter 10) and

$$\Delta(y, f(x)) := \begin{cases} f(x) & \text{if } y = 1 \\ -f(x) & \text{if } y = 0. \end{cases} \quad (6)$$

MART’s output  $f(x)$  models the log-odds of `spam` vs `email`,

$$f(x) = \log \frac{P(1 \mid x)}{P(0 \mid x)},$$

which makes the interpretation of (6) as conformity score very natural. (MART is known [7] to give good results on the Spambase dataset.)

Figure 1 gives the calibration plots for the CCP and for 8 random splits of the data set into a training set of size 3600 and a test set of size 1001 and of the training set into 5 or 10 folds. There is a further source of randomness as the MART procedure is itself randomized. The functions plotted in Figure 1 map each significance level  $\epsilon$  to the percentage of erroneous predictions made by the set predictor  $\Gamma^\epsilon$  on the test set. Visually, the plots are well-calibrated (close to the bisector of the first quadrant).

As for the efficiency of the CCP, see Table 1, which gives some statistics for the confidence and credibility output by the ICP and the 5-fold and 10-fold CCP. The columns labelled “0” to “7” give the mean values of confidence and credibility over the test set for various values of

the seed for the R pseudorandom number generator. The column labelled “Average” gives the average

$$\bar{v} := \frac{1}{8} \sum_{i=0}^7 v_i$$

of all the 8 mean values (which we denote  $v_0, \dots, v_7$ ) for the seeds 0–7, and the column labelled “St. dev.” gives the standard unbiased estimate

$$\sqrt{\frac{1}{7} \sum_{i=0}^7 (v_i - \bar{v})^2}$$

of the standard deviation of the mean values computed from  $v_0, \dots, v_7$ . The biggest advantage of the CCP is in the stability of its confidence values: the standard deviation of the mean confidences is much less than that for the ICP. However, the CCP also gives higher confidence; to some degree this can be seen from the table, but the high variance of the ICP confidence masks it: e.g., for the first 100 seeds the average of the mean confidence for ICP is 99.16% (with the standard deviation of the mean confidences equal to 0.149%, corresponding to the standard deviation of 0.015% of the average mean confidence).

#### 4. CONCLUSION

At this time there are no theoretical results about the validity of cross-conformal predictors (like Proposition 1), and it is an interesting open problem to establish such results.

#### 5. REFERENCES

- [1] Vladimir Vovk, Alex Gammerman, and Glenn Shafer, *Algorithmic Learning in a Random World*, Springer, New York, 2005.
- [2] Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman, “Qualified predictions for large data sets in the case of pattern recognition,” in *Proceedings of the First International Conference on Machine Learning and Applications (ICMLA)*, Las Vegas, NV, 2002, pp. 159–163, CSREA Press.
- [3] Anonymous, “Generalized conformal prediction for functional data,” Submitted to NIPS 2012, June 2012.
- [4] Vladimir Vovk, “Cross-conformal predictors,” Tech. Rep. arXiv:1208.0806v1 [stat.ML], arXiv.org e-Print archive, August 2012.
- [5] Jing Lei and Larry Wasserman, “Distribution free prediction bands,” Tech. Rep. arXiv:1203.5422 [stat.ME], arXiv.org e-Print archive, March 2012.
- [6] Anonymous, “Inductive conformal predictors in the batch mode,” Submitted to ACML 2012, July 2012.
- [7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, second edition, 2009.

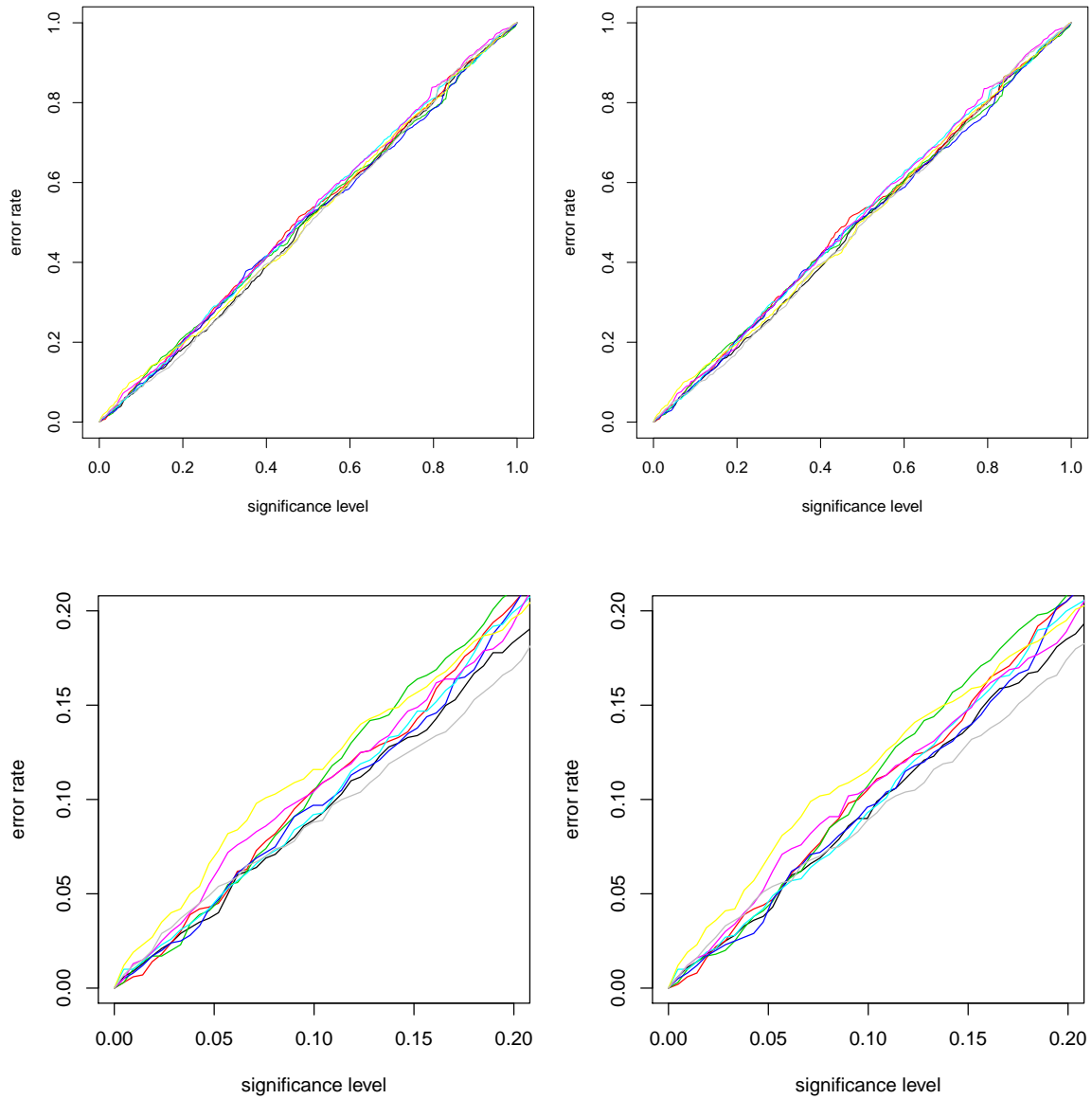


Figure 1. Top panels: the calibration plots for the cross-conformal predictor with  $K = 5$  (left) and  $K = 10$  (right) folds and the first 8 seeds, 0–7, for the R pseudorandom number generator. Bottom panels: the lower left corner of the corresponding top panel (which is the most important part of the calibration plot in applications).

Seed	0	1	2	3	4	5	6	7	Average	St. dev.
mean conf., ICP	99.25%	99.23%	99.00%	99.17%	99.30%	99.12%	99.38%	99.25%	99.21%	0.116%
mean cred., ICP	51.31%	50.37%	49.93%	52.45%	48.98%	50.34%	50.18%	52.00%	50.69%	1.148%
mean conf., $K = 5$	99.22%	99.17%	99.17%	99.24%	99.27%	99.27%	99.30%	99.30%	99.24%	0.054%
mean cred., $K = 5$	51.11%	49.74%	50.34%	50.69%	49.85%	49.49%	50.95%	51.46%	50.45%	0.713%
mean conf., $K = 10$	99.24%	99.20%	99.20%	99.23%	99.26%	99.28%	99.34%	99.32%	99.26%	0.051%
mean cred., $K = 10$	51.08%	49.74%	50.29%	50.77%	49.75%	49.48%	50.96%	51.45%	50.44%	0.727%

Table 1. Mean (over the test set) confidence and credibility for the ICP and the 5-fold and 10-fold CCP. The results are given for various values of the seed for the R pseudorandom number generator; column “Average” gives the average of all the 8 values for the seeds 0–7, and column “St. dev.” gives the standard unbiased estimate of the standard deviation computed from those 8 values.

# INFORMATION-THEORETIC METHODS FOR ANALYSIS AND INFERENCE IN ETYMOLOGY

Hannes Wettig<sup>1</sup>, Javad Nouri<sup>1</sup>, Kirill Reshetnikov<sup>2</sup> and Roman Yangarber<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Helsinki, Finland, First.Last@cs.helsinki.fi

<sup>2</sup>Academy of Sciences, Institute of Linguistics, Moscow, Russia.

## ABSTRACT

We introduce a family of minimum description length models which explicitly utilizes phonetic features and captures long-range contextual rules that condition recurrent correspondences of sounds within a language family. We also provide an algorithm to learn a model from this family given a corpus of cognates, sets of genetically related words. Finally, we present an *imputation* procedure which allows us to compare the quality of alignment models, as well as the goodness of the data sets. Our evaluations demonstrate that the new model yields improvements in performance, as compared to those previously reported in the literature.

## 1. INTRODUCTION

This paper introduces a family of context-aware models for alignment and analysis of etymological data on the level of phonetic features. We focus on discovering the rules of regular (or recurrent) phonetic correspondence across languages and determining genetic relations among a group of languages, based on linguistic data. In this work, we use the StarLing database of Uralic, [1], based on [2], restricted to the Finno-Ugric sub-family, consisting of 1898 *cognate sets*, as well as *Suomen Sanojen Alkuperä* (SSA), “The Origin of Finnish Words,” a Finnish etymological dictionary, [3], which contains over 5000 cognate sets. Elements within a given cognate set are words posited by the database creators to be derived from a common origin, a word-form in the ancestral *proto-language*.

One traditional arrangement of the Uralic languages—adapted from Encyclopedia Britannica—is shown in Figure 1; alternative arrangements found in the literature include moving Mari into a separate branch, or grouping it with Mordva into a branch, called “Volgaic”.

We aim to find the best *alignment* at the level of single sounds. The database itself only contains unaligned sets of corresponding words, with no notion of which sounds correspond, i.e., how the sounds align. We learn rules of phonetic correspondence allowing only the data to determine what rules underly it, using no externally supplied (and possibly biased) prior assumptions or “universal” principles—e.g., no preference to align vowel with vowels, a symbol with itself, etc. Therefore, all rules we find are *inherently encoded* in the corpus itself.

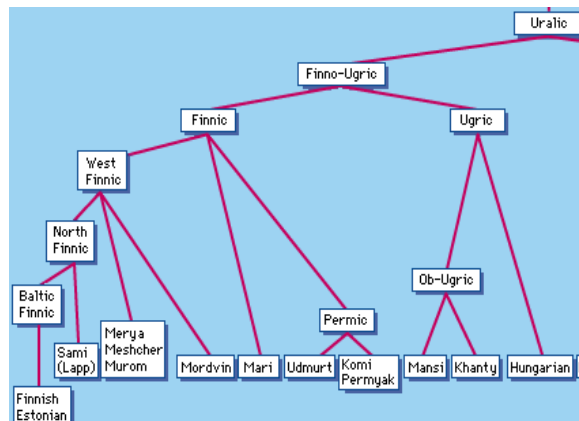


Figure 1. Finno-Ugric branch of Uralic language family

The criterion we use to choose a model (class) from the family we define is the code-length needed to communicate the complete (aligned) data. The learned minimum description length (MDL) models provide the desired alignments on the sound level, but also the underlying rules of correspondence, which enable us to *compress* the data. Apart from looking at the code-length, we also evaluate our models using an imputation (reconstruction of held-out data) procedure and by building phylogenies (family trees). We release the suite of etymological software for public use.

Most closely related to this work is our own previous work, e.g., [4], and work conducted at Berkeley, e.g., [5, 6]. The main improvement over these lies in awareness of a broader phonetic context of our models. We build decision trees to capture this context, where irrelevant context does not increase model complexity.

## 2. ALIGNING PAIRS OF WORDS

We begin with pairwise alignment: aligning pairs of words, from two related languages in our corpus of cognates. For each word pair, the task of alignment means finding exactly which symbols correspond. The simplest form of such alignment at the symbol level is a pair  $(\sigma : \tau) \in \Sigma \times T$ , a single symbol  $\sigma$  from the source alphabet  $\Sigma$  with a symbol  $\tau$  from the target alphabet  $T$ . We denote the sizes of the alphabets by  $|\Sigma|$  and  $|T|$ .

To model *insertions* and *deletions*, we augment both

alphabets with a special empty symbol—denoted by a dot—and write the augmented alphabets as  $\Sigma$  and  $T$ . We can then align word pairs such as *vuosi—al* (meaning “year” in Finnish and Xanty), for example as any of:

*v u o s i v u o s i* etc...  
 | | | | | | | | |  
*a l . . . . a . l .*

The alignment on the right then consists of the symbol pairs: (v:.), (u:a), (o:.), (s:l), (i:.).

### 3. FEATURE-WISE CONTEXT MODELS

Rather than encoding symbols (sounds) as atomic, we code them in terms of their phonetic features. To this end, the corpus has been transcribed into feature vectors, where each sound is represented as a vector of five multinomials, taking on two to eight values, where the first entry is its type (consonant or vowel) and the remaining four entries are as listed in Figure 2. We also encode word boundaries (denoted by #) and dots (deletions/insertions) as extra types, with no additional features.

Consonant articulation		
<b>M</b>	<b>Manner</b>	plosive, fricative, glide, ...
<b>P</b>	<b>Place</b>	labial, dental, ..., velar, uvular
<b>X</b>	<b>Voiced</b>	-, +
<b>S</b>	<b>Secondary</b>	-, affricate, aspirate, ...
Vowel articulation		
<b>V</b>	<b>Vertical</b>	high—mid—low
<b>H</b>	<b>Horizontal</b>	front—center—back
<b>R</b>	<b>Rounding</b>	-, +
<b>L</b>	<b>Length</b>	1—5

Figure 2. Phonetic features for consonants and vowels.

We employ the MDL Principle [7] for model class selection and the MDL cost consists of two parts. First, we encode the model class  $\mathcal{C}$ , which is determined by a set of 18 decision trees, one for each feature (type plus four consonant and four vowel features) on both levels—source and target language. These trees query some *context* at each inner node, and their leaves provide the distribution to be used to encode the corresponding feature of a sound. More precisely the model (class) is allowed to query a fixed, finite a set of *candidate* contexts. A context is a triplet  $(L, P, F)$ , where  $L$  is the level (source or target),  $P$  is a *position* relative to what we are currently encoding, and  $F$  is one of the possible features found at that position. An example of allowed candidate positions is given in Figure 3. In this setup, we have 2 levels  $\times$  8 positions  $\times$  2–6

Context Positions	
I	itself, possibly dot
-P	previous position, possibly dot
-S	previous non-dot symbol
-K	previous consonant
-V	previous vowel
+S	previous or self non-dot symbol
+K	previous or self consonant
+V	previous or self vowel
...	(other contexts possible)

Figure 3. Context positions that a feature tree may query.

features  $\approx 80$  candidate contexts, one of which defines an inner node of a feature tree. We can therefore encode each tree using one bit per node to indicate whether it is a leaf or not, plus about  $\log 80$  bits for each inner node to specify the context on which it splits. For a model class  $\mathcal{C}$ , we need to encode all of its 18 trees in this way, the resulting total code-length we denote  $L(\mathcal{C})$ .

The second part of the code-length comes from encoding the aligned data using model class  $\mathcal{C}$ . We encode the feature in some fixed order, type first for it determines which other features need to be encoded. For each sound and each feature, we take a path from the root of the corresponding tree of  $\mathcal{C}$  to a leaf, following at each inner node the branch that corresponds to the current context which is being queried. For example, when encoding feature **X** (voicedness) of a symbol  $\sigma$  in the source language we may arrive at a node given by  $(L, P, F) = (\textit{target}, -K, \textit{M})$  querying the manner of articulation of the previous consonant on the target level. This value (any manner of articulation or ‘n/a’ if there is no consonant on the target level between the current position and the beginning of the word) determines the edge we follow down the tree.

Each path from the root of a tree to a low-entropy leaf can be interpreted as a rule of phonetic correspondence. The path describes a contextual condition, the leaf gives the correspondence itself. High-entropy leaves represent variation that the model cannot explain.

In this way, all features of all symbols arrive at some node in the corresponding tree. We encode this data at each leaf independent of all other leaves using the normalized maximum likelihood (NML) distribution [8]. As the data at each leaf is multinomial, with cardinality  $|F|$ —the number of values feature  $F$  can take on—the corresponding code-length can be computed in linear time [9].

When  $\mathcal{C} = \{\mathcal{T}_F^L\}$  consists of trees  $\mathcal{T}_F^L$  for level  $L$  and feature  $F$ , and  $\mathcal{D}$  is the aligned corpus such that  $\mathcal{D}_{|L,F,\ell}$  is the portion arriving at a leaf  $\ell \in \mathcal{T}_F^L$ , then the overall code-length for  $\mathcal{D}$  using  $\mathcal{C}$  is

$$L(\mathcal{D}, \mathcal{C}) = L(\mathcal{C}) + \sum_L \sum_F \sum_\ell L_{NML}(\mathcal{D}_{|L,F,\ell}). \quad (1)$$

As implied,  $L_{NML}(\mathcal{D}_{|L,F,\ell})$  is the multinomial stochastic complexity of the restricted data  $\mathcal{D}_{|L,F,\ell}$ . This code-length is the criterion to be minimized by the learning algorithm.

### 4. LEARNING

We start with an initial *random* alignment for each pair of words in the corpus. We then alternatively re-build the decision trees for all features on source and target levels as described below, and re-align all word pairs in the corpus using standard dynamic-programming, an analog procedure to the one described in [4]. Both of these operations decrease code-length. We continue until we reach convergence.

Given a complete alignment of the data, for each level  $L$  and feature  $F$  we need to build a decision tree. We

want to minimize the MDL criterion (1), the overall code-length. We do so in a greedy fashion by iteratively splitting the level-feature restricted data  $\mathcal{D}_{|L,F}$  according to the cost-optimal decision (context to split upon). We start out by storing  $\mathcal{D}_{|L,F}$  at the root node of the tree, e.g., for the voicedness feature  $\mathbf{X}$  in Estonian (aligned to Finnish) we store data with counts:

+	801
-	821

In this example, there are 1622 occurrences of Estonian consonants in the data, 801 of which are voiced. The best split the algorithm found was on (Source, I,  $\mathbf{X}$ ), resulting in three new children. The data now splits according to this context into three subsets with counts:

+		-		n/a	
+	615	+	135	+	51
-	2	-	764	-	55

For each of these new nodes we split further, until no further drop in total code-length can be achieved. A split costs about  $\log 80$  plus the number of decision branches in bits, the achieved gain is the drop in the sum of stochastic complexities at the leaves obtained by splitting the data.

## 5. EVALUATION

We present two views on evaluation: a *strict* view and an *intuitive* view. From a strictly information-theoretic point of view, a sufficient condition to claim that model (class)  $M_1$  is better than  $M_2$ , is that  $M_1$  yields better compression of the data. Figure 4 shows the absolute costs (in bits) for

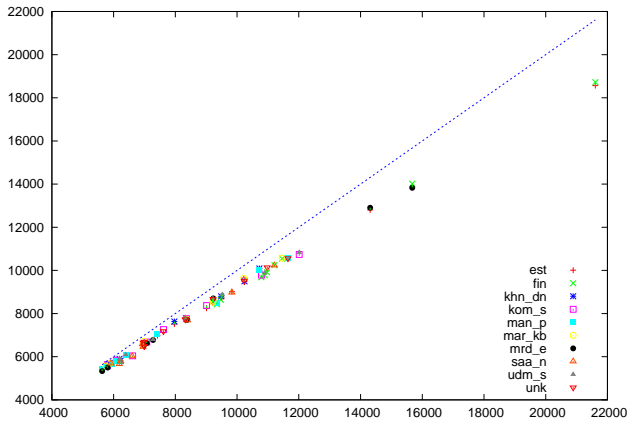


Figure 4. Comparison of code-lengths achieved by context model (Y-axis) and 1-1 baseline model (X-axis).

all language pairs<sup>1</sup>. The context model always has lower cost than the 1-1 baseline presented in [4]. In figure 5, we compare the context model against standard data compressors, Gzip and Bzip, as well as models from [4], tested on over 3200 Finnish-Estonian word pairs from SSA [3]. Gzip and Bzip need not encode any alignment, but neither can they exploit correspondence of sounds. These com-

<sup>1</sup>The labels appearing in the figures for the 10 Uralic languages used in the experiments are: est=Estonian, fin=Finnish, khn=Khanty, kom=Komi, man=Mansi, mar=Mari, mrd=Mordva, saa=Saami, udm=Udmurt, unk/ugr=Hungarian.

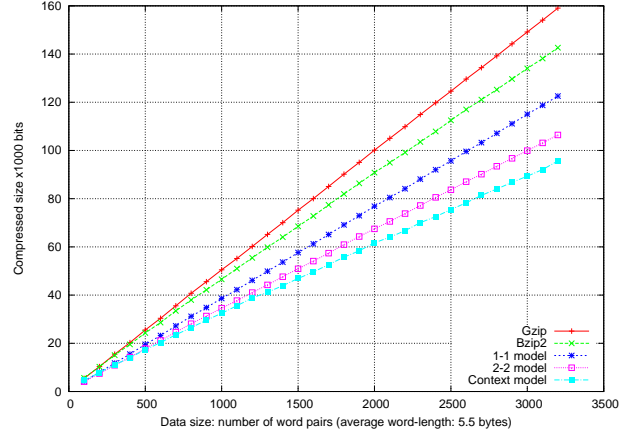


Figure 5. Comparison of compression power

parisons confirm that the new model finds more regularity in the data than the baseline model does, or an off-the-shelf data compressor, which has no knowledge that the words in the data are etymologically related.

For a more intuitive evaluation of the improvement in the model quality, we can compare the models by using them to *impute* unseen data. For a given model, and a language pair  $(L_1, L_2)$ —e.g., (Finnish, Estonian)—hold out one word pair, and train the model on the remaining data. Then show the model the hidden Finnish word and let it impute (i.e., guess) the corresponding Estonian. Imputation can be done for all models with a simple dynamic programming algorithm, very similar to the one used in the learning phase. Formally, given the hidden Finnish string, the imputation procedure selects from all possible Estonian strings the most probable Estonian string, given the model. Finally, we compute an edit distance (e.g., the Levenshtein distance) between the imputed string and the correct withheld Estonian word. We repeat this procedure for all word pairs in the  $(L_1, L_2)$  data set, sum the edit distances, and normalize by the total size (number of sounds) of the correct  $L_2$  data—giving the *Normalized Edit Distance*:  $NED(L_2|L_1, M)$  from  $L_1$  to  $L_2$ , under model  $M$ .

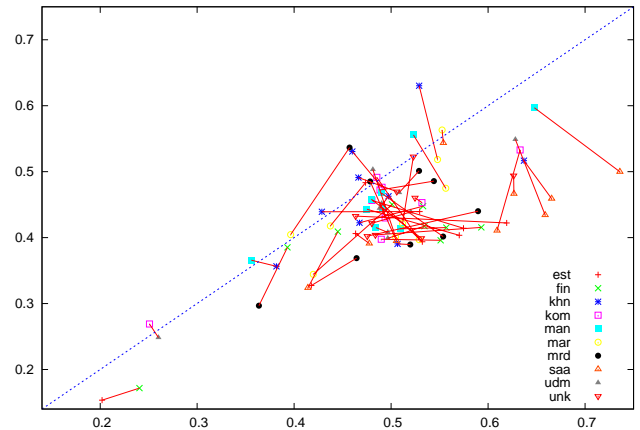


Figure 6. Comparison of NED of context model (Y-axis) and “two-part 1-1” model (X-axis).



The NED indicates how much regularity the model has captured. We use NED to compare models across all languages, Figure 6 compares the context model to the “two-part 1-1” model from [4]. Each of the  $10 \cdot 9$  points is a directed comparison of the two models: the source language is indicated in the legend, and the target language is identified by the other endpoint of the segment on which the point lies. The further away a point is from the diagonal, the greater the advantage of one model over the other.

The context model always has lower cost than the baseline, and lower NED in 88% of the language pairs. This is an encouraging indication that optimizing the code length is a good approach—the models do *not* optimize NED directly, and yet the cost correlates with NED, which is a simple and intuitive measure of model quality.

A similar use of imputation was presented in [5] as a kind of cross-validation. However, the novel, normalized NED measure we introduce here provides yet another inter-language distance measure (similarly to how NCD was used in [4]). The NED (distances) can be used to make inferences about how far the languages are from each other, via algorithms for drawing phylogenetic trees. The pairwise NED scores were fed into the NeighborJoin algorithm, to produce the phylogeny shown in Fig. 7.

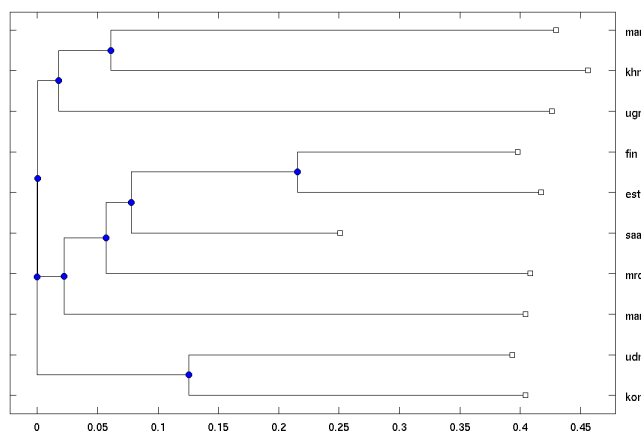


Figure 7. Finno-Ugric tree induced by imputation and normalized edit distances (via NeighborJoin)

To compare how far this is from a “gold-standard”, we can use, for example, a distance measure for unrooted, leaf-labeled (URLL) trees found in [10]. The URLL distance between this tree and the tree shown in Fig. 1 is 0.12, which is quite small. Comparison with a tree in which Mari is not coupled with either Mordva or Permic—which is currently favored in the literature on Uralic linguistics—makes it a perfect match.

## 6. DISCUSSION AND FUTURE WORK

We have presented a feature-based context-aware MDL alignment method and compared it against earlier models, both in terms of compression cost and imputation power. Language distances induced by imputation allow building of phylogenies. The algorithm takes only an etymological

data set as input, and requires no further assumptions. In this regard, it is as objective as possible, given the data (the data set itself, of course, may be highly subjective).

To our knowledge, this work represents a first attempt to capture *longer-range* context in etymological modeling, where prior work admitted minimum surrounding context for conditioning the edit rules or correspondences.

## Acknowledgments

This research was supported by the Uralink Project of the Academy of Finland, and by the National Centre of Excellence “Algorithmic Data Analysis (ALGODAN)” of the Academy of Finland. Suvi Hiltunen implemented earlier versions of the models.

## 7. REFERENCES

- [1] Sergei A. Starostin, “Tower of Babel: Etymological databases,” <http://newstar.rinet.ru/>, 2005.
- [2] Károly Rédei, *Uralisches etymologisches Wörterbuch*, Harrassowitz, Wiesbaden, 1988–1991.
- [3] Erkki Itkonen and Ulla-Maija Kulonen, *Suomen Sanojen Alkuperä (The Origin of Finnish Words)*, Suomalaisen Kirjallisuuden Seura, Helsinki, Finland, 2000.
- [4] Hannes Wettig, Suvi Hiltunen, and Roman Yangarber, “MDL-based Models for Alignment of Etymological Data,” in *Proceedings of RANLP: the 8th Conference on Recent Advances in Natural Language Processing*, Hissar, Bulgaria, 2011.
- [5] Alexandre Bouchard-Côté, Percy Liang, Thomas Griffiths, and Dan Klein, “A probabilistic approach to diachronic phonology,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, June 2007, pp. 887–896.
- [6] David Hall and Dan Klein, “Large-scale cognate recovery,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2011.
- [7] Peter Grünwald, *The Minimum Description Length Principle*, MIT Press, 2007.
- [8] Jorma Rissanen, “Fisher information and stochastic complexity,” *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40–47, January 1996.
- [9] Petri Kontkanen and Petri Myllymäki, “A linear-time algorithm for computing the multinomial stochastic complexity,” *Information Processing Letters*, vol. 103, no. 6, pp. 227–233, 2007.
- [10] D.F. Robinson and L.R. Foulds, “Comparison of phylogenetic trees,” *Math. Biosci.*, vol. 53, pp. 131–147, 1981.



# INFORMATION THEORETIC MODELS OF NATURAL LANGUAGE

*Łukasz Dębowski*

Institute of Computer Science, Polish Academy of Sciences,  
ul. Jana Kazimierza 5, 01-248 Warszawa, POLAND, ldebowsk@ipipan.waw.pl

## ABSTRACT

The relaxed Hilberg conjecture is a proposition about natural language which states that mutual information between two adjacent blocks of text grows according to a power law in function of the block length. In the paper two mathematical results connected to this conjecture are reviewed. First, we exhibit an example of a stochastic process, called the Santa Fe process, which is motivated linguistically and for which the mutual information grows according to a power law. Second, we demonstrate that a power law growth of mutual information implies a power law growth of vocabulary. The latter statement is observed for texts in natural language and called Herdan's law.

## 1. INTRODUCTION

It is often assumed that texts in natural language may be modeled by a stationary process and the entropy a random text can be determined [1]. More specifically, in 1990, German telecommunications engineer Wolfgang Hilberg conjectured that the entropy of a random text in natural language satisfies

$$H(X_1^n) \propto n^\beta, \quad (1)$$

where  $X_i$  are characters of the random text,  $X_n^m = (X_n, X_{n+1}, \dots, X_m)$  are blocks of consecutive characters,  $H(X) = \mathbf{E}[-\log P(X)]$  is the entropy of a discrete variable  $X$ , and  $\beta \in (0, 1)$  [2]. Hilberg's conjecture was based on an extrapolation of Shannon's seminal experimental data [3], which contained the estimates of conditional entropy for blocks of  $n \leq 100$  characters.

Statement (1) implies that the entropy rate  $h = \lim_{n \rightarrow \infty} H(X_1^n)/n$  equals 0. This in turn implies asymptotic determinism of utterances, which does not sound plausible. A more plausible modification of statement (1) is

$$I(X_1^n; X_{n+1}^{2n}) \propto n^\beta, \quad (2)$$

where  $I(X; Y) = H(X) + H(Y) - H(X, Y)$  is the mutual information between variables  $X$  and  $Y$ . We notice that relationship (2) arises for entropy

$$H(X_1^n) = An^\beta + hn \quad (3)$$

where  $h$  can be positive. Relationship (2) will be called the relaxed Hilberg conjecture.

In this paper, we will review some previous results of ours that concern two issues:

1. We exhibit an example of a stochastic process, called Santa Fe process, which is motivated linguistically and which satisfies relationship (2) asymptotically [4].
2. We demonstrate that relationship (2) implies that the text of length  $n$  contains at least  $n^\beta / \log n$  different words, under a certain plausible definition of a word [5]. Indeed, the power-law growth of the vocabulary is empirically observed for texts in natural language and called Herdan's law [6].

In our opinion, these results shed some light on probabilistic modeling of natural language.

## 2. THE SANTA FE PROCESS

Processes that satisfy the relaxed Hilberg conjecture arise in a very simple setting that resembles what may actually happen in natural language. Suppose that each statement  $X_i$  of a text in natural language can be represented as a pair  $X_i = (k, z)$  which states that the  $k$ -th proposition in some abstract enumeration assumes Boolean value  $z$ . Moreover, suppose that there is a stochastic process  $(K_i)_{i \in \mathbb{Z}}$  and a random field  $(Z_{ik})_{i \in \mathbb{Z}, k \in \mathbb{N}}$  such that if  $X_i = (k, z)$  then  $K_i = k$  and  $Z_{ik} = z$ . The process  $(K_i)_{i \in \mathbb{Z}}$  will be called the selection process and the field  $(Z_{ik})_{i \in \mathbb{Z}, k \in \mathbb{N}}$  will be called the object described by text  $(X_i)_{i \in \mathbb{Z}}$ . Note that variable  $Z_{ik}$  has two indices—the first one refers to the while  $i$  at which the statement  $X_i$  is made whereas the second one refers to the proposition  $K_i = k$ , which is either asserted or negated. Observe that statements that are made in texts fall under two types:

1. Statements about objects  $Z_{ik} = Z_k$ , which do not change in time (like mathematical or physical constants).
2. Statements about objects  $Z_{ik} \neq Z_{i+1,k}$ , which evolve with a varied speed (like culture, language, or geography).

We will obtain a power-law growth of mutual information for an appropriate choice of the selection process and the described object, namely, when the bits of the described object do not evolve too fast in comparison to their selection by the selection process.

In particular, the Santa Fe process  $(X_i)_{i \in \mathbb{Z}}$  will be defined as a sequence of random statements

$$X_i = (K_i, Z_{i, K_i}), \quad (4)$$

where processes  $(K_i)_{i \in \mathbb{Z}}$  and  $(Z_{ik})_{i \in \mathbb{Z}}$  with  $k \in \mathbb{N}$  are independent and distributed as follows. First, variables  $K_i$  are distributed according to the power law

$$P(K_i = k) = k^{-1/\beta} / \zeta(\beta^{-1}), \quad (K_i)_{i \in \mathbb{Z}} \sim \text{IID}, \quad (5)$$

where  $\beta \in (0, 1)$  and  $\zeta(x) = \sum_{k=1}^{\infty} k^{-x}$  is the zeta function. Second, each process  $(Z_{ik})_{i \in \mathbb{Z}}$  is a Markov chain with the marginal distribution

$$P(Z_{ik} = 0) = P(Z_{ik} = 1) = 1/2 \quad (6)$$

and the cross-over probabilities

$$P(Z_{ik} = 0 | Z_{i-1, k} = 1) = P(Z_{ik} = 1 | Z_{i-1, k} = 0) = p_k. \quad (7)$$

The name ‘‘Santa Fe process’’ has been chosen since the author discovered this process during a stay at the Santa Fe Institute.

Observe that the description given by the Santa Fe process is strictly repetitive for  $p_k = 0$ : if two statements  $X_i = (k, z)$  and  $X_j = (k', z')$  describe bits of the same address ( $k = k'$ ) then they always assert the same bit value ( $z = z'$ ). In this case the Santa Fe process is nonergodic. For strictly positive  $p_k$  the description is no longer strictly repetitive and the Santa Fe process is mixing [4].

By the following result, the Santa Fe process satisfies relationship (2) asymptotically:

**Theorem 1 ([4])** *Suppose  $\lim_{k \rightarrow \infty} p_k / P(K_i = k) = 0$ . Then the mutual information for the Santa Fe process obeys*

$$\lim_{n \rightarrow \infty} \frac{I(X_1^n; X_{n+1}^{2n})}{n^\beta} = \frac{(2 - 2^\beta)\Gamma(1 - \beta)}{[\zeta(\beta^{-1})]^\beta}. \quad (8)$$

Some processes over a finite alphabet which also satisfy relationship (2) asymptotically can be constructed by stationary coding of the Santa Fe process [4].

### 3. VOCABULARY GROWTH

In the second turn, we will show that the relaxed Hilberg conjecture can be related to the number of distinct words appearing in texts. It has been observed that words in natural language texts correspond in a good approximation to nonterminal symbols in the shortest grammar-based encoding of those texts [7, 8, 9]. Complementing this observation, we will demonstrate that relationship (2) constrains the number of distinct nonterminal symbols in the shortest grammar-based encoding of the random text.

A short introduction to grammar-based coding is in need. Briefly speaking, grammar-based codes compress strings by transforming them first into special grammars, called admissible grammars [10], and then encoding the grammars back into strings according to a fixed simple

method. An admissible grammar is a context-free grammar that generates a singleton language  $\{w\}$  for some string  $w \in \mathbb{X}^*$  [10]. In an admissible grammar, there is exactly one rule per nonterminal symbol and the nonterminals can be ordered so that the symbols are rewritten onto strings of strictly succeeding symbols [10]. Hence, such a grammar is given by its set of production rules

$$\left\{ \begin{array}{l} A_1 \rightarrow \alpha_1, \\ A_2 \rightarrow \alpha_2, \\ \dots, \\ A_n \rightarrow \alpha_n \end{array} \right\}, \quad (9)$$

where  $A_1$  is the start symbol, other  $A_i$  are secondary nonterminals, and the right-hand sides of rules satisfy  $\alpha_i \in (\{A_{i+1}, A_{i+2}, \dots, A_n\} \cup \mathbb{X})^*$ .

An example of an admissible grammar is

$$\left\{ \begin{array}{l} A_1 \mapsto A_2 A_2 A_4 A_5 \text{dear\_children} A_5 A_3 \text{all}. \\ A_2 \mapsto A_3 \text{you} A_5 \\ A_3 \mapsto A_4 \text{\_to\_} \\ A_4 \mapsto \text{Good\_morning} \\ A_5 \mapsto \text{\_} \end{array} \right\},$$

with the start symbol  $A_1$ , which produces the song

Good morning to you,  
Good morning to you,  
Good morning, dear children,  
Good morning to all.

For the shortest grammar-based encoding of a longer text in natural language, secondary nonterminals  $A_i$  often match the word boundaries, especially if it is required that these nonterminals are defined using only terminal symbols [9].

In the following,  $\mathbf{V}(w)$  will denote the number of distinct nonterminal symbols in the shortest grammar-based encoding of a text  $w$ . (The exact definition of the shortest grammar-based encoding, called admissibly minimal, is given in [5].) To connect the mutual information with  $\mathbf{V}(w)$ , we introduce another quantity, namely the length of the longest nonoverlapping repeat in a text  $w$ :

$$\mathbf{L}(w) := \max \{ |s| : w = x_1 s y_1 = x_2 s y_2 \wedge x_1 \neq x_2 \}, \quad (10)$$

where  $s, x_i, y_i \in \mathbb{X}^*$ . Using this concept, for processes over a finite alphabet we obtain this proposition.

**Theorem 2 ([5])** *Let  $(X_i)_{i \in \mathbb{Z}}$  be a stationary process over a finite alphabet. Assume that inequality*

$$\liminf_{n \rightarrow \infty} \frac{I(X_1^n; X_{n+1}^{2n})}{n^\beta} > 0 \quad (11)$$

*holds for some  $\beta \in (0, 1)$  and*

$$\sup_{n \geq 2} \mathbf{E} \left( \frac{\mathbf{L}(X_1^n)}{f(n)} \right)^q < \infty, \quad \forall q > 0, \quad (12)$$

*holds for some function  $f(n)$ . Then we have*

$$\limsup_{n \rightarrow \infty} \mathbf{E} \left( \frac{\mathbf{V}(X_1^n)}{n^\beta f(n)^{-1}} \right)^p > 0, \quad \forall p > 1. \quad (13)$$

An example of a process that satisfies the hypothesis of Theorem 2 with  $f(n) = \log n$  can be constructed by stationary coding of the Santa Fe process [11, 4]. However, for texts in natural language we have checked that there holds an empirical law  $\mathbf{L}(X_1^n) \approx \log^\alpha n$ , where  $\alpha \approx 2 \div 3$  [12]. It is an interesting open question how to construct processes which satisfy both (11) and  $\mathbf{L}(X_1^n) \approx \log^\alpha n$ .

#### 4. CONCLUSION

We have discussed some constructions and theorems for discrete-valued processes with long memory. Our results have very natural linguistic interpretations. We believe that the Santa Fe process deserves further investigation.

#### 5. REFERENCES

- [1] Thomas M. Cover and Roger C. King, “A convergent gambling estimate of the entropy of English,” *IEEE Trans. Inform. Theor.*, vol. 24, pp. 413–421, 1978.
- [2] Wolfgang Hilberg, “Der bekannte Grenzwert der redundanzfreien Information in Texten — eine Fehlinterpretation der Shannonschen Experimente?,” *Frequenz*, vol. 44, pp. 243–248, 1990.
- [3] Claude Shannon, “Prediction and entropy of printed English,” *Bell Syst. Tech. J.*, vol. 30, pp. 50–64, 1951.
- [4] Łukasz Dębowski, “Mixing, ergodic, and nonergodic processes with rapidly growing information between blocks,” *IEEE Trans. Inform. Theor.*, vol. 58, pp. 3392–3401, 2012.
- [5] Łukasz Dębowski, “On the vocabulary of grammar-based codes and the logical consistency of texts,” *IEEE Trans. Inform. Theor.*, vol. 57, pp. 4589–4599, 2011.
- [6] Gustav Herdan, *Quantitative Linguistics*, London: Butterworths, 1964.
- [7] J. Gerard Wolff, “Language acquisition and the discovery of phrase structure,” *Lang. Speech*, vol. 23, pp. 255–269, 1980.
- [8] Carl G. de Marcken, *Unsupervised Language Acquisition*, Ph.D. thesis, Massachusetts Institute of Technology, 1996.
- [9] Chunyu Kit and Yorick Wilks, “Unsupervised learning of word boundary with description length gain,” in *Proceedings of the Computational Natural Language Learning ACL Workshop, Bergen*, M. Osborne and E. T. K. Sang, Eds., pp. 1–6. 1999.
- [10] John C. Kieffer and Enhui Yang, “Grammar-based codes: A new class of universal lossless source codes,” *IEEE Trans. Inform. Theor.*, vol. 46, pp. 737–754, 2000.
- [11] Łukasz Dębowski, “Variable-length coding of two-sided asymptotically mean stationary measures,” *J. Theor. Probab.*, vol. 23, pp. 237–256, 2010.
- [12] Łukasz Dębowski, “Maximal lengths of repeat in English prose,” in *Synergetic Linguistics. Text and Language as Dynamic System*, Sven Naumann, Peter Grzybek, Relja Vulcanović, and Gabriel Altmann, Eds., pp. 23–30. Wien: Praesens Verlag, 2012.

# INFORMATION-THEORETIC PROBABILITY COMBINATION WITH APPLICATIONS TO RECONCILING STATISTICAL METHODS

*David R. Bickel, University of Ottawa*

## 1. MOTIVATION

The analysis of biological data often requires choices between methods that seem equally applicable and yet that can yield very different results. This occurs not only with the notorious problems in frequentist statistics of conditioning on one of multiple ancillary statistics and in Bayesian statistics of selecting one of many appropriate priors, but also in choices between frequentist and Bayesian methods, in whether to use a potentially powerful parametric test to analyze a small sample of unknown distribution, in whether and how to adjust for multiple testing, and in whether to use a frequentist model averaging procedure. Today, statisticians simultaneously testing thousands of hypotheses must often decide whether to apply a multiple comparisons procedure using the assumption that the p-value is uniform under the null hypothesis (theoretical null distribution) or a null distribution estimated from the data (empirical null distribution). While the empirical null reduces estimation bias in many situations [1], it also increases variance [2] and can substantially increase bias when the data distributions have heavy tails [3]. Without any strong indication of which method can be expected to perform better for a particular data set, combining their estimated false discovery rates or adjusted p-values may be the safest approach.

Emphasizing the reference class problem, [4] pointed out the need for ways to assess the evidence in the diversity of statistical inferences that can be drawn from the same data. Previ-

ous applications of p-value combination have included combining inferences from different ancillary statistics [5], combining inferences from more robust procedures with those from procedures with stronger assumptions, and combining inferences from different alternative distributions [6]. However, those combination procedures are only justified by a heuristic Bayesian argument and have not been widely adopted. To offer a viable alternative, the problem of combining conflicting methods is framed herein in terms of probability combination.

Most existing methods of automatically combining probability distributions have been designed for the integration of expert opinions. For example, [7], [8], and [9] proposed combining distributions to minimize a weighted sum of Kullback-Leibler divergences from the distributions being combined, with the weights determined subjectively, e.g., by the elicitation of the opinions of the experts who provided the distributions or by the extent to which each expert is considered credible. Under broad conditions, that approach leads to the linear combination of the distributions that is defined by those weights [7, 9].

Such *linear opinion pools* also result from this *marginalization property*: any linearly combined marginal distribution is the same whether marginalization or combination is carried out first [10]. The marginalization property forbids certain counterintuitive combinations of distributions, including any combination of distributions that differs in a probability assignment from the unanimous assignment of all distributions combined [11, p. 173]. Combinations violating the marginal-

ization property can be expected to perform poorly as estimators regardless of their appeal as distributions of belief. On the other hand, invariance to reversing the order of Bayesian updating and distribution combination instead requires a *logarithmic opinion pool*, which uses a geometric mean in place the arithmetic mean of the linear opinion pool; see, e.g., [12, §4.11.1] or [13]. While that property is preferable to the marginalization property from the point of view of a Bayesian agent making decisions on the basis of independent reports of other Bayesian agents, it is less suitable for combining distributions that are highly dependent or that are distribution estimates rather than actual distributions of belief.

## 2. GAME-THEORETIC FRAMEWORK

Like the opinion pools of Section 1, the strategy introduced in [14] is intended for combining distributions based on the same data or information as opposed to combining distributions based on independent data sets. However, to relax the requirement that the distributions be provided by experts, the weights are optimized rather than specified. While the new strategy leads to a linear combination of distributions, the combination hedges by including only the most extreme distributions rather than all of the distributions. In addition, the game leading to the hedging takes into account any known constraints on the true distribution. (This game is distinct from those of [15, 16], which apply [17] to blending frequentist and Bayesian statistical methods.)

The game that generates the hedging strategy is played between three players: the mechanism that generates the true distribution (“Nature”), a statistician who never combines distributions (“Chooser”), and a statistician who is willing to combine distributions (“Combiner”). Nature must select a distribution that complies with constraints known to the statisticians, who want to choose distributions as close as possible to the distribution chosen by Nature. Other things being equal, each statistician would also like to select a distribution that is as much better than that of the other statistician as possible. Thus, each statistician seeks primarily to

come close to the truth and secondarily to improve upon the distribution selected by the other statistician. Combiner has the advantage over Chooser that the former may select any distribution, whereas the latter must select one from a given set of the distributions that estimate the true distribution or that encode expert opinion. On the other hand, Combiner is disadvantaged in that the game rules specify that Nature seeks to maximize the gain of Chooser albeit without concern for the gain of Combiner. Since Nature favors Chooser without opposing Combiner, the optimal strategy of Combiner is one of hedging but is less cautious than the minimax strategies that are often optimal for typical two-player zero-sum games against Nature. The distribution chosen according to the strategy of Combiner will be considered the combination of the distributions available to Chooser. The combination distribution is a function not only of the combining distributions but also of the constraints on the true distribution.

[14] encodes the game and strategy described above in terms of Kullback-Leibler loss and presents its optimal solution as a general method of combining distributions. The special case of combining discrete distributions is summarized in the next section. A framework for using the proposed combination method to resolve method conflicts in point and interval estimation, hypothesis testing, and other aspects of statistical data analysis appear in [14] with an application to the combination of three false discovery rate methods for the analysis of microarray data.

## 3. SPECIAL CASE: COMBINING DISCRETE DISTRIBUTIONS

Let  $\mathcal{P}$  denote the set of probability distributions on  $(\Xi, 2^\Xi)$ , where  $\Xi$  is a finite set. It is written as  $\Xi = \{0, 1, \dots, |\Xi| - 1\}$  without loss of generality. Then the information divergence of  $P \in \mathcal{P}$  with respect to  $Q \in \mathcal{P}$  reduces to

$$D(P||Q) = \sum_{i \in \Xi} P(\{i\}) \log \frac{P(\{i\})}{Q(\{i\})}.$$

For any  $P \in \mathcal{P}$  and the random variable  $\xi$  of distribution  $P$ , the  $|\Xi|$ -tuple

$$T(P) = (P(\xi = 0), P(\xi = 1), \dots, P(\xi = |\Xi| - 1))$$

will be called the *tuple representing*  $P$ .

Consider  $\mathcal{P}^* = \{P_\phi : \phi \in \Phi\}$ , a nonempty subset of  $\mathcal{P}$ . Every  $\phi \in \Phi$  corresponds to a different random variable and thus to a different  $|\Xi|$ -tuple.

**Lemma.** *Let  $\mathcal{P}^*$  denote a nonempty, finite subset of  $\mathcal{P}$ , and let  $\text{ext } \mathcal{P}^*$  denote the set of distributions that are represented by the extreme points of the convex hull of the set of tuples representing the members of  $\mathcal{P}^*$ . If there are a  $Q \in \mathcal{P}$  and a  $C > 0$  such that  $D(P^*||Q) = C$  for all  $P^* \in \text{ext } \mathcal{P}^*$ , then  $Q$  is the centroid of  $\mathcal{P}^*$ .*

*Proof.* As an immediate consequence of what [18] labels “Theorem (Csiszár)” and “Theorem 1,”

$$\min_{P'' \in \mathcal{P}} \max_{P' \in \mathcal{P}^*} D(P' || P'') = C.$$

By definition, the centroid is the solution of that *minimax redundancy* problem.  $\square$

The **Theorem** in [14] that connects the lemma to the following corollary is based on the *redundancy-capacity theorem*, the celebrated relationship between capacity and minimax redundancy. The redundancy-capacity theorem was presented by R. G. Gallager in 1974 [19, Editor’s Note] and published as [20] and [21]; cf. [22]. [23, Theorem 13.1.1], [24, §5.2.1], and [25, Problem 8.1] provide useful introductions. The extension from discrete distributions to general probability measures ([26]; [27]) is exploited in [14].

The combination of a set of probabilities of the same hypothesis or event is simply the linear combination or mixture of the highest and lowest of the plausible probabilities in the set such that the mixing proportion is optimal:

**Corollary.** *Let  $P^+$  denote the combination of the distributions in  $\check{\mathcal{P}} \subseteq \mathcal{P}$  with truth constrained by  $\dot{\mathcal{P}} \subseteq \mathcal{P}$ . Suppose  $c$  distributions on  $(\{0, 1\}, 2^{\{0,1\}})$  are to be combined  $(\check{\mathcal{P}} = \{\check{P}_1, \dots, \check{P}_c\})$ , and let  $\mathfrak{P}_0 = \{\dot{P}(\{0\}) : \dot{P} \in \dot{\mathcal{P}}\}$  and  $\underline{\check{P}}, \bar{\check{P}} \in \mathcal{P}$  such that  $\underline{\check{P}}(\{0\}) = \min \check{P}_i(\{0\})$  and  $\bar{\check{P}}(\{0\}) = \max \check{P}_i(\{0\})$ . If there is at least one  $i \in \{1, \dots, c\}$  for which  $\check{P}_i(\{0\}) \in \mathfrak{P}_0$  holds, then  $P^+ = w^+ \underline{\check{P}} + (1 - w^+) \bar{\check{P}}$ , where  $w^+ =$*

$$\arg \sup_{w \in [0,1]} \left( w \Delta(\underline{\check{P}} || w) + (1 - w) \Delta(\bar{\check{P}} || w) \right);$$

$$\Delta(\bullet || w) = D\left(\bullet || w \underline{\check{P}} + (1 - w) \bar{\check{P}}\right).$$

#### 4. ACKNOWLEDGMENTS

Most of this extended abstract is derived from [14] with permission from Elsevier.

#### 5. REFERENCES

- [1] Bradley Efron, “Size, power and false discovery rates,” *Annals of Statistics*, vol. 35, pp. 1351–1377, 2007.
- [2] B. Efron, *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, Cambridge University Press, Cambridge, 2010.
- [3] D. R. Bickel, “Estimating the null distribution to adjust observed confidence levels for genome-scale screening,” *Biometrics*, vol. 67, pp. 363–370, 2011.
- [4] Ole E. Barndorff-Nielsen, “Diversity of evidence and Birnbaum’s theorem,” *Scandinavian Journal of Statistics*, vol. 22, pp. 513–515, 1995.
- [5] IJ Good, “A Bayesian interpretation of ancillarity,” *Journal of Statistical Computation and Simulation*, vol. 19, no. 4, pp. 302–308, 1984.
- [6] I. J. Good, “Significance tests in parallel and in series,” *Journal of the American Statistical Association*, vol. 53, pp. 799–813, 1958.
- [7] M Toda, “Information-receiving behavior of man,” *Psychological Review*, vol. 63, pp. 204–212, 1956.
- [8] A. E. Abbas, “A Kullback-Leibler View of Linear and Log-Linear Pools,” *Decision Analysis*, vol. 6, pp. 25–37, 2009.
- [9] Jan Kracík, “Combining marginal probability distributions via minimization of weighted sum of Kullback-Leibler divergences,” *International Journal of Approximate Reasoning*, vol. 52, pp. 659–671, 2011.

- [10] K. J. McConway, "Marginalization and linear opinion pools," *Journal of the American Statistical Association*, vol. 76, pp. 410–414, 1981.
- [11] Roger M. Cooke, *Experts in Uncertainty: Opinion and Subjective Probability in Science*, Oxford University Press, 1991.
- [12] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer, New York, 1985.
- [13] R. T. Clemen and R. L. Winkler, "Combining probability distributions from experts in risk analysis," *Risk Analysis*, vol. 19, pp. 187–203, 1999.
- [14] D. R. Bickel, "Game-theoretic probability combination with applications to resolving conflicts between statistical methods," *International Journal of Approximate Reasoning*, vol. 53, pp. 880–891, 2012.
- [15] D. R. Bickel, "Controlling the degree of caution in statistical inference with the Bayesian and frequentist approaches as opposite extremes," *Electron. J. Statist.*, vol. 6, pp. 686–709, 2012.
- [16] D. R. Bickel, "Blending Bayesian and frequentist methods according to the precision of prior information with applications to hypothesis testing," *Working Paper, University of Ottawa, deposited in uO Research at <http://hdl.handle.net/10393/23124>*, 2012.
- [17] Flemming Topsøe, "Information theoretical optimization techniques," *Kybernetika*, vol. 15, no. 1, pp. 8–27, 1979.
- [18] K. Nakagawa and F. Kanaya, "A new geometric capacity characterization of a discrete memoryless channel," *IEEE Transactions on Information Theory*, vol. 34, pp. 318–321, 1988.
- [19] B. Ryabko, "Comments on 'A source matching approach to finding minimax codes' by Davisson, L. D. and Leon-Garcia, A.," *IEEE Transactions on Information Theory*, vol. 27, pp. 780–781, 1981.
- [20] B.Y. Ryabko, "Encoding of a source with unknown but ordered probabilities," *Prob. Pered. Inform.*, vol. 15, pp. 71–77, 1979.
- [21] L. Davisson and a. Leon-Garcia, "A source matching approach to finding minimax codes," *IEEE Transactions on Information Theory*, vol. 26, pp. 166–174, 1980.
- [22] Robert G. Gallager, "Source coding with side information and universal coding," *Technical Report LIDS-P-937, Laboratory for Information Decision Systems, MIT*, 1979.
- [23] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, 2006.
- [24] J. Rissanen, *Information and Complexity in Statistical Modeling*, Springer, New York, 2007.
- [25] Imre Csiszár and János Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Cambridge University Press, Cambridge, 2011.
- [26] D Haussler, "A general minimax result for relative entropy," *IEEE Transactions on Information Theory*, vol. 43, pp. 1276 – 1280, 1997.
- [27] P.D. Grünwald and A. P. Dawid, "Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory," *Annals of Statistics*, vol. 32, pp. 1367–1433, 2004.

## Information-Theoretic Value of Evidence Analysis Using Probabilistic Expert Systems

Anjali Mazumder (EDMI Services Ltd., Warwick (from 01/09/12)), Steffen Lauritzen (Oxford)

The evaluation and interpretation of evidence is often made under uncertainty where the task of reasoning involves estimating unknown quantities from some given observations. There is often a quest for data to reduce uncertainty. Forensic scientists are often called upon in courts to give expert testimony in a court of law or public enquiry, e.g. the source of a DNA sample. Their evaluation and interpretation of the evidence is often under scrutiny and they are often asked to justify their decision-making process. The task of decision-making, evaluating, and interpreting the evidence is further tested when there are multiple sources of evidence which may or may not relate to the same query. Information is seldom cost free and therefore there is a need to evaluate beforehand whether it is worthwhile to acquire and to decide which (sources of information) to consult that would optimise the desired goal (i.e. reduction in uncertainty about the inference).

Using information-theoretic concepts, Lauritzen and Mazumder (2008) defined a value of evidence (VOE) criterion  $I_q$  as a general measure of informativeness for any forensic query  $Q$  and collection of evidence  $X_1, \dots, X_K$  where the probability distribution of the query (given evidence) is of interest. When there are multiple sources of information, a decision-theoretic framework provides a systematic approach to considering which test(s) to perform that most contributes to reducing uncertainty regarding the query. A probabilistic network formulation provides an attractive platform for the graph-theoretic representation of the VOE problem and eases the laborious calculation of marginal and conditional probabilities of interest. When the configuration space for exact computations and exhaustive searching is infeasible, Monte Carlo sampling methods are employed.

The VOE criterion  $I_q$ , having a solid theoretical basis, has been directly applied to a variety of planning problems in forensic genetics to determine the quantity and choice of individuals and genetic markers to type to gain sufficient information for evaluation and interpretation (Mazumder, 2010). This approach is extended to consider other complex evidential reasoning cases involving multiple evidence types in which the graph modular structure and conditional independence properties are exploited to aid the decision-making and reasoning process. This research aims to contribute in three ways: (1) developing computational methods in VOE analysis using PESs, (2) developing a decision-theoretic framework for planning and inference in the evaluation of complex evidence structures, and advancing the evaluation and interpretation of forensic evidence methods.

### *References*

Lauritzen, S. and Mazumder, A. (2008). Informativeness of genetic markers for forensic inference - an information-theoretic approach. *Forensic Science International: Genetics Supplement Series*, 1:652-653.

Mazumder, A. (2010). *Planning in Forensic DNA Identification Using Probabilistic Expert Systems*, Doctoral dissertation, University of Oxford, Oxford.



# MDL-BASED IDENTIFICATION OF RELEVANT TEMPORAL SCALES IN TIME SERIES

Ugo Vespier<sup>1</sup>, Arno Knobbe<sup>1</sup>, Siegfried Nijssen<sup>2</sup> and Joaquin Vanschoren<sup>1</sup>

<sup>1</sup>LIACS, Leiden University, the Netherlands

<sup>2</sup>Katholieke Universiteit Leuven, Belgium

uvespier@liacs.nl

## ABSTRACT

The behavior of many complex physical systems is affected by a variety of phenomena occurring at different temporal scales. Time series data produced by measuring properties of such systems often mirrors this fact by appearing as a composition of signals across different time scales. When the final goal of the analysis is to model the individual phenomena affecting a system, it is crucial to be able to recognize the right temporal scales and to separate the individual components of the data. We introduce a solution to this challenge based on a combination of the Minimum Description Length (MDL) principle, feature selection strategies, and convolution techniques from the signal processing field. As a result, we show that our algorithm produces a good decomposition of a given time series and, as a side effect, builds a compact representation of its identified components.

## 1. INTRODUCTION

Our work [5] is concerned with the analysis of sensor data. When monitoring complex physical systems over time, one often finds multiple phenomena in the data that work on different time scales. If one is interested in analyzing and modeling these individual phenomena, it is crucial to recognize these different scales and separate the data into its underlying components. Here, we present a method for extracting the time scales of various phenomena present in large time series.

The need for analyzing time series data at multiple time scales is nicely demonstrated by a large monitoring project in the Netherlands, called *InfraWatch* [4]. In this project, we employ a range of sensors to measure the dynamic response of a large Dutch highway bridge to varying traffic and weather conditions. When viewing this data (see Fig. 1, upper plot), one can easily distinguish various *transient events* in the signal that occur on different time scales. Most notable are the gradual change in strain over the course of the day (as a function of the outside temperature, which influences stiffness parameters of the concrete), a prolonged increase in strain caused by rush hour traffic congestion, and individual bumps in the signal due to cars and trucks traveling over the bridge. In order to understand the various changes in the sensor signal, one would benefit substantially from separating out the events at various scales. The main goal of the work described

here is to do just that: we consider the temporal data as a series of superimposed effects at different time scales, establish at which scales events most often occur, and from this we extract the underlying signal components.

We approach the scale selection problem from a Minimum Description Length [1] (MDL) perspective. The motivation for this is that we need a framework in which we can deal with a wide variety of representations for scale components. Our main assumption is that separating the original signal into components at different time scales will simplify the shape of the individual components, making it easier to model them separately. Our results show that, indeed, these multiple models outperform (in terms of MDL score) a single model derived from the original signal. While introducing multiple models incurs the penalty of having to describe them, there are much fewer ‘exceptions’ to be described compared to the single model, yielding a lower overall description length.

The analysis of time scales in time series data is often approached from a *scale-space* perspective, which involves convolution of the original signal with Gaussian kernels of increasing size [6] to remove information at smaller scales. By subtracting carefully selected components of the scale-space, we can effectively cut up the scale space into  $k$  ranges. In other words, signal processing offers methods for producing a large collection of derived features, and the challenge we face in this paper is how to select a subset of  $k$  features, such that the original signal is decomposed into a set of meaningful components at different scales.

Our approach applies the MDL philosophy to various aspects of modeling: choosing the appropriate scales at which to model the components, determining the optimal number of components (while avoiding overfitting on overly specific details of the data), and deciding which class of models to apply to each individual component. For this last decision, we propose two classes of models representing the components respectively on the basis of a discretization and a segmentation scheme. For this last scheme, we allow three levels of complexity to approximate the segments: piecewise constant approximations, piecewise linear approximations, as well as quadratic ones. These options result in different trade-offs between model cost and accuracy, depending on the type of signal we are dealing with.

A useful side product of our approach is that it identifies a concise representation of the original signal. This representation is useful in itself: queries run on the decomposed signal may be answered more quickly than when run on the original data. Furthermore, the parameters of the encoding may indicate useful properties of the data as well.

## 2. PRELIMINARIES

We deal with finite sequences of numerical measurements (samples), collected by observing some property of a system with a sensor, and represented in the form of time series as defined below.

**Definition 1.** A *time series* of length  $n$  is a finite sequence of values  $\mathbf{x} = x[1], \dots, x[n]$  of finite precision.<sup>1</sup> A *subsequence*  $\mathbf{x}[a : b]$  of  $\mathbf{x}$  is defined as follows:

$$\mathbf{x}[a : b] = (\mathbf{x}[a], \mathbf{x}[a + 1], \dots, \mathbf{x}[b]), \quad a < b$$

We also assume that all the considered time series have no missing values and that their sampling rate is constant.

### 2.1. The Scale-Space Image

The *scale-space image* [6] is a scale parametrization technique for one-dimensional signals<sup>2</sup> based on the operation of convolution.

**Definition 2.** Given a signal  $\mathbf{x}$  of length  $n$  and a response function (kernel)  $\mathbf{h}$  of length  $m$ , the result of the *convolution*  $\mathbf{x} * \mathbf{h}$  is the signal  $\mathbf{y}$  of length  $n$ , defined as:

$$y[t] = \sum_{j=-m/2+1}^{m/2} x[t-j] h[j]$$

In this paper,  $\mathbf{h}$  is a Gaussian kernel with mean  $\mu = 0$ , standard deviation  $\sigma$ , area under the curve equal to 1, discretized into  $m$  values.<sup>3</sup>

Given a signal  $\mathbf{x}$ , the family of  $\sigma$ -smoothed signals  $\Phi_{\mathbf{x}}$  over scale parameter  $\sigma$  is defined as follows:

$$\Phi_{\mathbf{x}}(\sigma) = \mathbf{x} * \mathbf{g}_{\sigma}, \quad \sigma > 0$$

where  $\mathbf{g}_{\sigma}$  is a Gaussian kernel having standard deviation  $\sigma$ , and  $\Phi_{\mathbf{x}}(0) = \mathbf{x}$ .

The signals in  $\Phi_{\mathbf{x}}$  define a surface in the time-scale plane  $(t, \sigma)$  known in the literature as the *scale-space image* [3, 6]. This visualization gives a complete description of the scale properties of a signal in terms of Gaussian smoothing. For practical purposes, the scale-space image is quantized across the scale dimension by computing the convolutions only for a finite number of scale parameters. More formally, for a given signal  $\mathbf{x}$ , we fix a set of scale parameters  $S = \{2^i \mid 0 \leq i \leq \sigma_{max} \wedge i \in \mathbb{N}\}$  and we compute  $\Phi_{\mathbf{x}}(\sigma)$  only for  $\sigma \in S$  where  $\sigma_{max}$  is such that  $\Phi_{\mathbf{x}}(\sigma)$  is approximately equal to the mean signal of  $\mathbf{x}$ .

<sup>1</sup>32-bit floating point values in our experiments.

<sup>2</sup>From now on, we will use the term signal and time series interchangeably.

<sup>3</sup>To capture almost all non-zero values, we define  $m = \lfloor 6\sigma \rfloor$ .

## 2.2. Scale-Space Decomposition

We define a decomposition scheme of a signal  $\mathbf{x}$  by considering adjacent ranges of scales of the signal scale-space image as below.

**Definition 3.** Given a signal  $\mathbf{x}$  and a set of  $k - 1$  scale parameters  $C = \{\sigma_1, \dots, \sigma_{k-1}\}$  (called the *cut-point set*) such that  $\sigma_1 < \dots < \sigma_{k-1}$ , the *scale decomposition* of  $\mathbf{x}$  is given by the set of component signals  $D_{\mathbf{x}}(C) = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ , defined as follows:

$$\mathbf{x}_i = \begin{cases} \Phi_{\mathbf{x}}(0) - \Phi_{\mathbf{x}}(\sigma_1) & \text{if } i = 1 \\ \Phi_{\mathbf{x}}(\sigma_{i-1}) - \Phi_{\mathbf{x}}(\sigma_i) & \text{if } 1 < i < k \\ \Phi_{\mathbf{x}}(\sigma_{k-1}) & \text{if } i = k \end{cases}$$

Note that for  $k$  components we require  $k - 1$  cut-points.

## 3. MDL SCALE DECOMPOSITION SELECTION

Given an input signal  $\mathbf{x}$ , the main computational challenge we face is twofold:

- find a good subset of cut-points  $C$  such that the resulting  $k$  components of the decomposition  $D_{\mathbf{x}}(C)$  optimally capture the effect of transient events at different scales,
- select a representation for each component, according to its inherent complexity.

We propose to use the Minimum Description Length (MDL) principle to approach this challenge. The two-part MDL principle states that the best model  $M$  to describe the signal  $\mathbf{x}$  is the one that minimizes the sum of the description lengths  $L(M) + L(\mathbf{x} \mid M)$ .

The possible models depend on the scale decomposition  $D_{\mathbf{x}}(C)$  considered<sup>4</sup> and on the representations used for its individual components. An ideal set of representations would adapt to the specific features of every single component, resulting in a concise summarization of the decomposition and, thus, of the signal. In order to apply the MDL principle, we need to define a model  $M_{D_{\mathbf{x}}(C)}$  for a given scale decomposition  $D_{\mathbf{x}}(C)$  and, consequently, how to compute both  $L(M_{D_{\mathbf{x}}(C)})$  and  $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$ . The latter term is the length in bits of the information lost by the model, i.e., the residual signal  $\mathbf{x} - M_{D_{\mathbf{x}}(C)}$ .

Note that, in order to employ MDL, we discretize the input signal  $\mathbf{x}$ . Below, we introduce the proposed representation schemes for the components. We also define the bit complexity of the residual and the model selection procedure.

### 3.1. Component Representation Schemes

Within our general framework, many different approaches could be used for representing the components of a decomposition. In the next paragraphs we introduce two such methods.

<sup>4</sup>Including the decomposition formed by zero cut-points ( $C = \emptyset$ ), i.e., the signal itself.

### 3.1.1. Discretization-based representation

As a first representation, we propose to consider more coarse-grained discretizations of the original range of values. By doing so, similar values will be grouped together in the same bin. The resulting sequence of integers is compacted further by performing run-length encoding, resulting in a string of  $(v, l)$  pairs, where  $l$  represents the number of times value  $v$  is repeated consecutively. This string is finally encoded using a Shannon-Fano or Huffman code (see Section 3.2).

### 3.1.2. Segmentation-based representation

The main assumption on which we base this method is that a clear transient event can be accurately represented by a simple function, such as a polynomial of a bounded degree. Hence, if a signal contains a number of clear transient events, it should be possible to accurately represent this signal with a number of segments, each of which represented by a simple function.

Given a component  $\mathbf{x}_i$  of length  $n$ , let

$$z(\mathbf{x}_i) = \{t_1, t_2, \dots, t_m\}, \quad 1 < t_i \leq n$$

be a set of indexes of the segment boundaries.

Let  $\text{fit}(\mathbf{x}_i[a : b], d_i)$  be the approximation of  $\mathbf{x}_i[a : b]$  obtained by fitting a polynomial of degree  $d_i$ . Then, we represent each component  $\mathbf{x}_i$  with the approximation  $\hat{\mathbf{x}}_i$ , such that:

$$\begin{aligned} \hat{\mathbf{x}}_i[0 : z_1] &= \text{fit}(\mathbf{x}_i[0 : z_1], d_i) \\ \hat{\mathbf{x}}_i[z_i : z_{i+1}] &= \text{fit}(\mathbf{x}_i[z_i : z_{i+1}], d_i), \quad 1 \leq i < m \\ \hat{\mathbf{x}}_i[z_m : n] &= \text{fit}(\mathbf{x}_i[z_m : n], d_i) \end{aligned}$$

Note that approximation  $\hat{\mathbf{x}}_i$  is quantized again by reapplying the function  $Q$  to each of its values.

For a given  $k$ -component scale decomposition  $D_{\mathbf{x}}(C)$  and a fixed polynomial degree for each of its components, we calculate the complexity in bits of the model  $M_{D_{\mathbf{x}}(C)}$ , based on this representation scheme, as follows. Each approximated component  $\hat{\mathbf{x}}_i$  consists of  $|z(\mathbf{x}_i)| + 1$  segments. For each segment, we need to represent its length and the  $d_i + 1$  coefficients of the fitted polynomial. The length  $ls_i$  of the longest segment in  $\hat{\mathbf{x}}_i$  is given by

$$ls_i = \max(z_1 \cup \{z_{i+1} - z_i \mid 0 < i \leq m\})$$

We therefore use  $\log_2(ls_i)$  bits to represent the segment lengths, while for the coefficients of the polynomials we employ floating point numbers of fixed<sup>5</sup> bit complexity  $c$ . The MDL model cost is thus defined, omitting minor terms, as:

$$L(M_{D_{\mathbf{x}}(C)}) = \sum_{i=1}^k (|z(\mathbf{x}_i)| + 1) (\lceil \log_2(ls_i) \rceil + c(d_i + 1))$$

So far we assumed to have a set of boundaries  $z(\mathbf{x}_i)$ , but we did not specify how to compute them. A desirable

<sup>5</sup>In our experiments  $c = 32$ .

property for our segmentation would be that a segmentation at a coarser scale does not contain more segments than a segmentation at a finer scale.

The scale space theory assures that there are fewer zero-crossing of the derivatives of a signal at coarser scales [6]. In our segmentation we use the zero-crossings of the first and second derivatives.

## 3.2. Residual Encoding

Given a model  $M_{D_{\mathbf{x}}(C)}$ , its residual  $\mathbf{r} = \mathbf{x} - \sum_{i=1}^k \hat{\mathbf{x}}_i$ , computed over the component approximations, represents the information of  $\mathbf{x}$  not captured by the model. Having already defined the model cost for the two proposed encoding schemes, we only still need to define  $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$ , i.e., a bit complexity  $L(\mathbf{r})$  for the residual  $\mathbf{r}$ .

Here, we exploit the fact that we operate in a quantized space; we encode each bin in the quantized space with a code that uses approximately  $-\log(P(x))$  bits, where  $P(x)$  is the frequency of the  $x$ th bin in our data. The main justification for this encoding is that we expect that the errors are normally distributed around 0. Hence, the bins in the discretization that reflect a low error will have the highest frequency of occurrences; we will give these the shortest codes. In practice, ignoring small details, such codes can be obtained by means of Shannon-Fano coding or Huffman coding; as Hu et al. [2] we use Huffman coding in our experiments.

## 3.3. Model Selection

We can now define the MDL score that we are optimizing as follows:

**Definition 4.** Given a model  $M_{D_{\mathbf{x}}(C)}$ , its **MDL score** is defined as:

$$L(M_{D_{\mathbf{x}}(C)}) + L(\mathbf{r})$$

In the case of discretization-based encoding, the MDL score is affected by the cardinality used to encode each component. In the case of segmentation-based encoding the MDL score depends on the boundaries of the segments and the degrees of the polynomials in the representation. In both cases, also the cut-points of the considered decomposition affect the final score.

The simplest way to find the model that minimizes this score is to enumerate, encode and compute the MDL score for every possible scale-space decomposition and all possible encoding parameters. This brute-force approach results to be feasible in practice.

## 4. EXPERIMENTS

In this section, we experimentally evaluate our method actual sensor data from a real-world application. For a complete evaluation of the method, including a more systematic one over artificial data, please refer to [5].

We consider the strain measurements produced by a sensor attached to a large highway bridge in the Netherlands. The considered time series consists of 24 hours of strain measurements sampled at 1 Hz (totaling 86,400

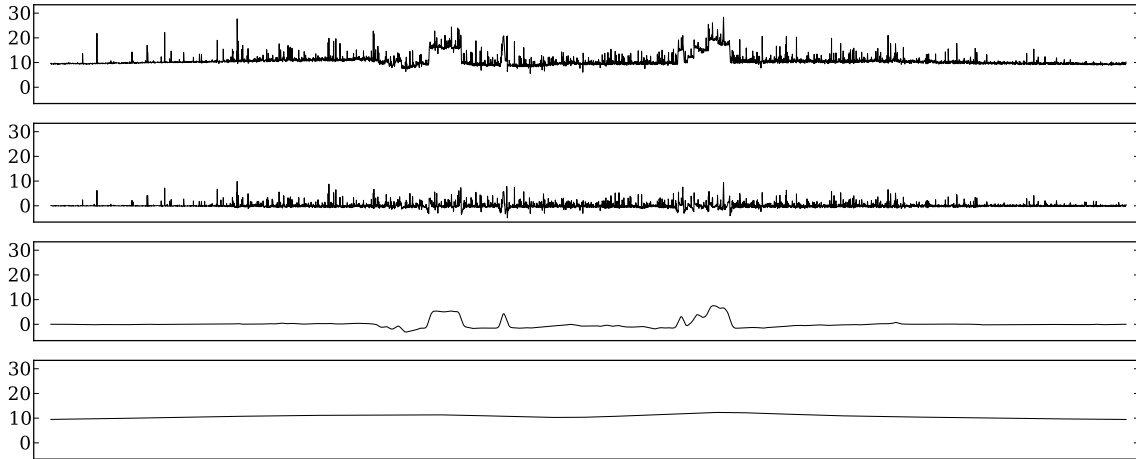


Figure 1: Signal (top) and top-ranked scale decomposition for the InfraWatch data.

data points). A plot of the data is shown in Figure 1 (top-most plot). We evaluated all the possible decompositions up to three components (two cut-points) allowing both the representation schemes we introduced. In the case of the discretization-based representations, we limit the possible cardinalities to 4, 16 and 64. The top-ranked decomposition results in 3 components as shown in the last three plots in Figure 1. The selected cut-points appear at scales  $2^6 = 64$  and  $2^{11} = 2048$ . All three components are represented with the discretization-based scheme, with a cardinality of respectively 4, 16, and 16 symbols. The decomposition has an MDL-score of 344, 276, where  $L(M) = 19, 457$  and  $L(D | M) = 324, 818$ . The found components accurately correspond to physical events on the bridge. The first component, covering scales lower than  $2^6$ , reflects the short-term influence caused by passing vehicles and represented as peaks in the signal. Note that the cardinality selected for this component is the lowest admissible in our setting (4). This is reasonable considering that the relatively simple dynamic behavior occurring at these scales, mostly the presence or not of a peak over a flat baseline, can be cheaply described with 4 or fewer states without incurring a too large error. The middle component, covering scales between  $2^6$  and  $2^{11}$ , reflects the medium-term effects caused by traffic jams. The first component is slightly influenced by the second one, especially at the start and ending points of a traffic jam. Finally, the third component captures all the scales greater than  $2^{11}$ , here representing the effect of temperature during a whole day. To sum up, the top-ranked decomposition successfully reflects the real physical phenomena affecting the data. The decompositions with rank 8 or less all present similar configurations of cut-points and cardinalities, resulting in comparable components where the conclusions above still hold. The first 2-component decomposition appears at rank 10 with the cut-point placed at scale  $2^6$ , which separates the short-term peaks from all the rest of the signal (traffic jams and baseline mixed together). These facts make the result pretty stable as most of the good decompositions are ranked first.

## 5. CONCLUSIONS AND FUTURE WORK

We introduced a novel methodology to discover the fundamental scale components in a time series in an unsupervised manner. The methodology is based on building candidate scale decompositions, defined over the scale-space image [6] of the original time series, with an MDL-based selection procedure aimed at choosing the optimal one.

As shown, our approach identifies the relevant scale components in a relevant real-world application, giving meaningful insights about the data.

Future work will experiment with diverse representation schemes and hybrid approaches (such as using combinations of segmentation, discretization and Fourier-based encodings).

## 6. REFERENCES

- [1] P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, 2007.
- [2] B. Hu, T. Rakthanmanon, Y. Hao, S. Evans, S. Lonardi, and E. Keogh. Discovering the intrinsic cardinality and dimensionality of time series using mdl. In *Proceedings of ICDM 2011*, pages 1086–1091, 2011.
- [3] T. Lindeberg. Scale-space for discrete signals. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 12(3):234–254, Mar. 1990.
- [4] U. Vespier *et al.* Traffic Events Modeling for Structural Health Monitoring. In *Proceedings IDA 2011*, 2011.
- [5] U. Vespier *et al.* MDL-based Analysis of Time Series at Multiple Time-Scales. In *Proceedings ECML-PKDD 2012*, 2012.
- [6] A. P. Witkin. Scale-space filtering. In *Proceedings IJCAI 1983*, pages 1019–1022, San Francisco, CA, USA, 1983.

# MODEL SELECTION FOR MULTIVARIATE STOCHASTIC PROCESSES

Jesús E. García<sup>1</sup>, Verónica A. González-López<sup>2</sup> and M. L. L. Viola<sup>3</sup>

<sup>1 2</sup> Department of statistics, University of Campinas,

Rua Sergio Buarque de Holanda, 651 Cidade Universitária CEP 13083-859 Campinas, SP, Brazil.

<sup>3</sup> Department of statistics, Universidade Federal de São Carlos,

Via Washington Luís, km 235 - Bairro Monjolinho CEP 13.565-905 - São Carlos, SP, Brazil.

## ABSTRACT

We address the problem of model selection for a multivariate source with finite alphabet. Families of Markov models and model selection algorithms are generalized for the multivariate case. For Markovian sources our model selection procedures are consistent in the sense that, eventually, as the collected data grows, the sources Markov model will be retrieved exactly and it will be described with a minimal number of parameters.

## 1. INTRODUCTION

Multivariate Markov chains are used for modeling stochastic processes arising on many areas as for example linguistics, biology and neuroscience. There are diverse models families from which to choose a model for a given data set. For example Markov chains of order  $m$ , variable length Markov chains (VLMC) see for example (5), (6), (2) or partition Markov models see (4). On each family, the selection of a specific Markov model gives information about the dependence structure for the dataset.

A recurrent problem is to model multiple streams of finite memory data with distributions that are suspected to be dependent or similar or equal. In the case of independent sources, the interest is to find the differences and similarities between the distribution of the sources. In the dependent case we want to find the dependence structure for the multivariate source. In this paper we propose a class of Markov models for each of that cases (dependent or independent sources), that generalize the partition Markov models for multivariate sources. We show procedures to, given a dataset, select a model in our class of models, that approximate the joint law of the source. The procedure are consistent in the sense that if the law of the source is Markovian, eventually, as the collected data grow, the source's Markov model will be retrieved exactly. This work extend and generalize previous results about minimal Markov models and context tree models as in (4), (6), (2), (1) and (3). In section 2 we revisit the family of partition Markov models. In section 3 we address the problem of simultaneously modeling multiple data sources. Finally in section 4 we show a procedure to estimate the internal structure of dependence between the coordinates of a multivariate stationary source.

## 2. MARKOV CHAIN WITH PARTITION $\mathcal{L}$

Let  $(X_t)$  be a discrete time, finite memory Markov chain on a finite alphabet  $A$ . Denote the string  $a_m a_{m+1} \dots a_n$  by  $a_m^n$ , where  $a_i \in A$ ,  $m \leq i \leq n$ . Let  $M$  be the maximum memory for the process, and  $S = A^M$ .

For each  $a \in A$  and  $s \in S$ ,

$$P(a|s) = \text{Prob}(X_t = a | X_{t-M}^{t-1} = s);$$

**Definition 2.1.** Let  $(X_t)$  be a discrete time order  $M$  Markov chain on a finite alphabet  $A$ . We will say that  $s, r \in S$  are equivalent (denoted by  $s \sim_p r$ ) if  $P(a|s) = P(a|r) \forall a \in A$ .

For any  $s \in S$ , the equivalence class of  $s$  is given by  $[s] = \{r \in S | r \sim_p s\}$ .

**Remark 2.1.** The equivalence relationship defines a partition of  $S$ . The parts of this partition are the equivalence classes. The classes are the subsets of  $S$  with the same transition probabilities i.e.  $s, r \in S$  belongs to different classes if and only if they have different transition probabilities.

**Remark 2.2.** We can think that each element of  $S$  on the same equivalence class activates the same random mechanism to choose the next element in the Markov chain.

We can define now the a Markov chain with partition  $\mathcal{L}$ .

**Definition 2.2.** let  $(X_t)$  be a discrete time, order  $M$  Markov chain on  $A$  and let  $\mathcal{L} = \{L_1, L_2, \dots, L_K\}$  be a partition of  $S$ . We will say that  $(X_t)$  is a Markov chain with partition  $\mathcal{L}$  if this partition is the one defined by the equivalence relationship  $\sim_p$  introduced by definition 2.1.

Let  $\mathcal{L} = \{L_1, L_2, \dots, L_K\}$  be the partition of  $(X_t)$

$$P(a|L_i) = P(a|s), \text{ for any } s \in L_i$$

**Remark 2.3.** The set of parameters for a Markov chain over the alphabet  $A$  with partition  $\mathcal{L}$  can be denoted by,

$$\{P(a|L) : a \in A, L \in \mathcal{L}\}.$$

If we know the equivalence relationship for a given Markov chain, then we need  $(|A| - 1)$  transition probabilities for each class to specify the model. Then the number of parameters for the model is  $|\mathcal{L}|(|A| - 1)$ .

## 2.1. Partition Markov model selection

Let  $x_1^n$  be a sample of the process  $(X_t)$ ,  $s \in \mathcal{S}$ ,  $a \in A$  and  $n > M$ . We denote by  $N_n(s, a)$  the number of occurrences of the string  $s$  followed by  $a$  in the sample  $x_1^n$ ,

$$N_n(s, a) = |\{t : M < t \leq n, x_{t-M}^{t-1} = s, x_t = a\}|, \quad (1)$$

the number of occurrences of  $s$  in the sample  $x_1^n$  is denoted by  $N_n(s)$  and

$$N_n(s) = |\{t : M < t \leq n, x_{t-M}^{t-1} = s\}|. \quad (2)$$

To simplify the notation we will omit the  $n$  on  $N_n$ .

## 2.2. A distance in $\mathcal{S}$

**Definition 2.3.** We define the distance  $d$  in  $\mathcal{S}$ ,

$$\begin{aligned} d(s, r) &= \frac{1}{\ln(n)} \sum_{a \in A} \left\{ N(s, a) \ln \left( \frac{N(s, a)}{N(s)} \right) \right. \\ &\quad + N(r, a) \ln \left( \frac{N(r, a)}{N(r)} \right) \\ &\quad \left. - (N(s, a) + N(r, a)) \ln \left( \frac{N(s, a) + N(r, a)}{N(s) + N(r)} \right) \right\} \end{aligned}$$

for any  $s, r \in \mathcal{S}$ .

**Proposition 2.1.** For any  $s, r \in \mathcal{S}$ ,

- i.  $d(s, r) \geq 0$  with equality if and only if  $\frac{N(s, a)}{N(s)} = \frac{N(r, a)}{N(r)} \quad \forall a \in A$ ,
- ii.  $d(s, r) = d(r, s)$ ,

**Remark 2.4.**  $d$  can be generalized to subsets (see (4)).

**Theorem 2.1.** (Consistence in the case of a Markov source) Let  $(X_t)$  be a discrete time, order  $M$  Markov chain on a finite alphabet  $A$ . Let  $x_1^n$  be a sample of the process, then for  $n$  large enough, for each  $s, r \in \mathcal{S}$ ,  $d(r, s) < \frac{(|A|-1)}{2}$  iff  $s$  and  $r$  belong to the same class.

**Algorithm 2.1.** (Partition selection algorithm)

**Input:**  $d(s, r) \forall s, r \in \mathcal{S}$ ; **Output:**  $\hat{\mathcal{L}}_n$ .

$B = \mathcal{S}$

$\hat{\mathcal{L}}_n = \emptyset$

**while**  $B \neq \emptyset$

**select**  $s \in B$

**define**  $L_s = \{s\}$

$B = B \setminus \{s\}$

**for each**  $r \in B, r \neq s$

**if**  $d(s, r) < \frac{(|A|-1)}{2}$

$L_s = L_s \cup \{r\}$

$B = B \setminus \{r\}$

$\hat{\mathcal{L}}_n = \hat{\mathcal{L}}_n \cup \{L_s\}$

**Return:**  $\hat{\mathcal{L}}_n = \{L_1, L_2, \dots, L_K\}$

If the source is Markovian, for  $n$  large enough, the algorithm returns the partition for the source.

**Corollary 2.1.** Under the assumptions of Theorem 2.1,  $\hat{\mathcal{L}}_n$ , given by the algorithm 2.1 converges almost surely eventually to  $\mathcal{L}^*$ , where  $\mathcal{L}^*$  is the partition of  $\mathcal{S}$  defined by the equivalence relationship.

## 3. GENERALIZED PARTITION MARKOV MODELS FOR MULTIPLE INDEPENDENT FINITE MEMORY SOURCES

In this section we extend the family of models for multiple independent sources of data. We also extend our algorithm. As in (4), the procedure is consistent and tight, for Markovian sources, eventually, as the data grow, the source's Markov model will be retrieved exactly and described with the minimal number of parameters.

We will consider a dataset which consist of  $K$  sequences of size  $n_k$ , for  $k = 1, \dots, K$ .

### 3.1. Model family

Let  $(X_t^k)$  for  $k = 1, \dots, K$  be the  $K$  independent finite memory stochastic processes, all of them stationary and ergodic. For each process  $(X_t^k)$  let  $S_k$  and  $d_k$  be the state space and order of the respective Markov model.

**Definition 3.1.**  $\mathcal{S} = \{(s, k) : s \in S_k, k = 1, 2, \dots, K\}$

For each  $a \in A$  and  $(s, k) \in \mathcal{S}$ ,

$$P_k(a|s) = \text{Prob}(X_t^k = a | X_{t-M}^{k, t-1} = s);$$

The models in our family are indexed by the partition defined in the following equivalence relation.

**Definition 3.2.** We will say that  $(s, i), (r, j) \in \mathcal{S}$  are equivalent (denoted by  $(s, i) \sim_{P, K} (r, j)$ ) if  $P_i(a|s) = P_j(a|r) \quad \forall a \in A$ . For any  $(s, i) \in \mathcal{S}$ , the equivalence class of  $(s, i)$  is given by  $[(s, i)] = \{(r, j) \in \mathcal{S} | (r, j) \sim_{P, K} (s, i)\}$ .

We can define now the a set of Markov chain with partition  $\mathcal{L}$ .

**Definition 3.3.** let  $X$  be a set of  $K$  independent Markov chains on  $A$  and let  $\mathcal{L} = \{L_1, L_2, \dots, L_K\}$  be a partition of  $\mathcal{S}$ . We will say that  $X$  is a set of Markov chains with partition  $\mathcal{L}$  if this partition is the one defined by the equivalence relationship  $\sim_{P, K}$  introduced by definition 3.2.

**Remark 3.1.** The parameters for a set of independent Markov chains over the alphabet  $A$  with partition  $\mathcal{L}$  is,

$$\{P(a|L) : a \in A, L \in \mathcal{L}\},$$

where  $P(a|L) = P_i(a|s)$  for any  $(i, s) \in L$ .

**The number of parameters for the model is  $|\mathcal{L}|(|A|-1)$ .**

### 3.2. A distance between sequences

**Definition 3.4.** For any  $(s, i), (r, j) \in \mathcal{S}$ , we define the distance  $d_K((s, i), (r, j))$  in  $\mathcal{S}$  as

$$\begin{aligned} d_K((s, i), (r, j)) &= \frac{1}{\ln(n)} \sum_{a \in A} \left\{ N_i(s, a) \ln \left( \frac{N_i(s, a)}{N_i(s)} \right) \right. \\ &+ N_j(r, a) \ln \left( \frac{N_j(r, a)}{N_j(r)} \right) \\ &- (N_i(s, a) + N_j(r, a)) \times \\ &\left. \times \ln \left( \frac{N_i(s, a) + N_j(r, a)}{N_i(s) + N_j(r)} \right) \right\}, \end{aligned}$$

where  $N_i(s)$  and  $N_i(s, a)$  are the number of times that the sequences  $s$  and  $sa$  respectively appear in the sample  $i$ .

**Proposition 3.1.**  $d_K(\cdot, \cdot)$  have the following properties,

- i.  $d_K((s, i), (r, j)) \geq 0$  with equality if and only if  $\frac{N_i(s, a)}{N_i(s)} = \frac{N_j(r, a)}{N_j(r)} \quad \forall a \in A$ ,
- ii.  $d_K((s, i), (r, j)) = d_K((r, j), (s, i))$ ,

To simplify the notation and without loss of generality we will suppose that all the sequences have the same size  $n$ .

**Theorem 3.1.** (Consistence in the case of Markov sources) Let  $X$  be a set of independent Markov chain of finite order;  $(x_1^{i, n})_{i=1}^K$  a size  $n$  sample of each process. For each  $(s, i), (r, j) \in \mathcal{S}$  for  $n$  large enough,  $d_K((s, i), (r, j)) < \frac{|A|-1}{2}$  iff  $(s, i)$  and  $(r, j)$  belong to the same class.

The same algorithm 2.1 can be used (with  $d_K(\cdot, \cdot)$ ) to estimate the partition for the set of chains.

## 4. MULTIVARIATE SOURCES

In this section we will consider the case in which we have a multivariate source with dependent coordinates.

To simplify the notation, we will assume that the partition Markov model is known. Our objective is to obtain for each part a partition of the set of coordinates on independent sets. The same procedure can be used to find subsets of the coordinates that are conditionally independent.

Let  $(X_t)$  be a Markov chain on  $A = B^l$  with partition  $\mathcal{L}$ . For  $U = \{u_1, \dots, u_k\} \subset \{1, 2, \dots, l\}$  and  $a = (a_1, \dots, a_l) \in A$ , define:

i)  $a^u = (a_{u_1}, \dots, a_{u_k})$ ,

ii) for any  $L \in \mathcal{L}$ ,

$$P(a^U | L) = \text{Prob}(X_t^U = a^U | X_{t-M}^{t-1} = s) \quad \forall s \in L,$$

iii) for  $s \in \mathcal{S}$

$$N_n(s, a^U) = |\{t : M < t \leq n, x_{t-M}^{t-1} = s, x_t^U = a^U\}|,$$

iv) for  $L \in \mathcal{L}$

$$N_n^{\mathcal{L}}(L, a^U) = \sum_{s \in L} N_n(s, a^U).$$

### Example

Consider  $B = \{0, 1, 2\}$  with dimension  $l = 2$ , the alphabet will be  $A = B^2 = \{0, 1, 2\}^2$ . For  $L \in \mathcal{L}$ , we need to specify  $P(a|L)$ , this means  $(|A| - 1) = 8$  parameters for each  $L$ . If for a fixed  $L$  the first coordinate is independent from the second then  $P(a|L) = P(a_1|L)P(a_2|L) \quad \forall a \in A$  and the number of parameter will be  $(|B| - 1) + (|B| - 1) = 4$  for this  $L$ .

In general, for  $A = B^l$ , fix  $L \in \mathcal{L}$  and a partition  $\mathcal{I}_L$  of  $\{1, 2, \dots, l\}$  in independent coordinates, we have that

$$P(a|L) = \prod_{C \in \mathcal{I}_L} P(a^C | L) \quad \forall a \in A$$

and the number of parameters needed for the part  $L$  will be

$$\sum_{C \in \mathcal{I}} (|B|^{|C|} - 1)$$

### 4.1. Conditional dependence structure

**Definition 4.1.** For each  $L \in \mathcal{L}$ , define  $\mathcal{I}_L$  as de maximal partition of  $\{1, 2, \dots, l\}$  such that

$$P(a|L) = \prod_{C \in \mathcal{I}_L} P(a^C | L) \quad \forall a \in A.$$

We will say that  $\mathcal{I}_{\mathcal{L}} = \{\mathcal{I}_L\}_{L \in \mathcal{L}}$  is the structure of conditional dependence for the process.

### 4.2. Estimating the conditional dependence structure

Our procedure to estimate  $\mathcal{I}_{\mathcal{L}}$  is based on the Bayesian information criterion (BIC).

$$P(x_1^n) = P(x_1^M) \prod_{L \in \mathcal{L}, a \in A} \prod_{C \in \mathcal{I}_L} P(a^C | L)^{N_n^{\mathcal{L}}(L, a)}.$$

The maxima for  $\prod_{L \in \mathcal{L}, a \in A} \prod_{C \in \mathcal{I}_L} P(a^C | L)^{N_n^{\mathcal{L}}(L, a)}$  is

$$\text{ML}(\mathcal{L}, \mathcal{I}_{\mathcal{L}}, x_1^n) = \prod_{L \in \mathcal{L}, a \in A} \prod_{C \in \mathcal{I}_L} \left( \frac{N_n^{\mathcal{L}}(L, a^C)}{N_n^{\mathcal{L}}(L)} \right)^{N_n^{\mathcal{L}}(L, a)},$$

and the BIC criterion for ou class of models,

$$\begin{aligned} \text{BIC}(\mathcal{L}, \mathcal{I}_{\mathcal{L}}, x_1^n) &= \ln(\text{ML}(\mathcal{L}, \mathcal{I}_{\mathcal{L}}, x_1^n)) \\ &- \sum_{L \in \mathcal{L}} \sum_{C \in \mathcal{I}_L} (|A|^{|C|} - 1) \frac{\ln(n)}{2}. \end{aligned}$$

For a Markovian source the BIC model selection methodology is consistent.

### 4.3. Consistence

**Theorem 4.1.** Let  $(X_t)$  be a Markov chain of order  $M$  over a finite alphabet  $A$ , with partition  $\mathcal{L}^*$  and structure of conditional dependence  $\mathcal{I}_{\mathcal{L}^*}$ . Define,

$$\mathcal{I}_{\mathcal{L}_n} = \arg \max_{\mathcal{I} \in \mathcal{D}} \{\text{BIC}(\mathcal{L}_n, \mathcal{I}, x_1^n)\},$$

Where  $\mathcal{D}$  is the set of all possible structures of dependences for  $A$  and  $\mathcal{L}_n$ , then, eventually almost surely as  $n \rightarrow \infty$ ,

$$\mathcal{I}_{\mathcal{L}^*} = \mathcal{I}_{\mathcal{L}_n}$$

The next Theorem shows that is not necessary to search for the maxima on  $\mathcal{D}$ .

Consider any collection of partitions of  $\{1, 2, \dots, l\}$ ,

$$\mathcal{D} = \{D_L\}_{L \in \mathcal{L}}.$$

Fix  $L_0 \in \mathcal{L}$  and  $U, V \in D_{L_0}, U \neq V$ . Define  $\mathcal{D}^{L_0, U, V}$  as the collection of partitions containing the same partitions than  $\mathcal{D}$  except  $D_{L_0}$  is substituted by

$$D_{L_0} \setminus \{\{U\}, \{V\}\} \cup \{U \cup V\}.$$

**Theorem 4.2.** *Let  $(X_t)$  be a Markov chain over  $A = B^l$  with partition  $\mathcal{L}$ , then,*

$$P(a^{U \cup V} | L_0) = P(a^U | L_0)P(a^V | L_0) \forall a \in A$$

*if, and only if, eventually almost surely as  $n \rightarrow \infty$ ,*

$$BIC(\mathcal{L}, \mathcal{D}^{L_0, U, V}, x_1^n) < BIC(\mathcal{L}, \mathcal{D}, x_1^n).$$

## 5. CONCLUSION

In this paper we study two generalizations of previous results about minimal Markov models to the multivariate case. First, we consider the case in which we have multiple independent sources. We model all the sources simultaneously and the model selection algorithm returns not only the set of equivalent states for each source, it also identify all the states in all sources which can be considered equivalents between them. In this way, even strings activating the same random mechanism on different sources are identified and classified. The second generalization correspond to a stationary source with a multivariate alphabet. In this case we first choose a partition Markov model and then, for the transition probabilities of each part, we identify the maximal partition of the set of coordinates such that the different parts are independent. A similar procedure and algorithm can be used to find subsets of coordinates which are conditionally independent.

## 6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support for this research provided by CNPq projects 485999/2007-2 and 476501/2009-1 and USP project “Mathematics, computation, language and the brain”.

## 7. REFERENCES

- [1] BUHLMANN P. and WYNER A. (1999). Variable length Markov chains. *Ann. Statist.* **27** 480–513.
- [2] CSISZÁR, I. and TALATA, Z. (2006). Context tree estimation for not necessarily finite memory processes, via BIC and MDL. *IEEE Trans. Inform. Theory* **52** 1007–1016.
- [3] GALVES, A., GALVES, C., GARCIA, J. E., GARCIA, N. L. and LEONARDI, F. (2012). Context tree selection and linguistic rhythm retrieval from written texts. *Annals of Applied Statistics*, **6** 1, 186 (2012).

- [4] GARCIA, J. and GONZALEZ-LOPEZ, V. (2010) Minimal Markov Models, arXiv:1002.0729v1.
- [5] RISSANEN J. (1983). A universal data compression system, *IEEE Trans. Inform. Theory* **29**(5) 656 – 664.
- [6] WEINBERGER, M., RISSANEN, J. and FEDER, M. (1995). A universal finite memory source, *IEEE Trans. Inform. Theory* **41**(3) 643 – 652.



# PENALIZED LEAST SQUARES MODEL AVERAGING

Erkki P. Liski<sup>1</sup> and Antti Liski<sup>2</sup>

<sup>1</sup>School of Information Sciences, University of Tampere,  
FIN-33014 Tampere, FINLAND, Erkki.Liski@uta.fi

<sup>2</sup>Institute of Signal Processing, Tampere University of Technology,  
P.O.Box 553, FIN-33101 Tampere, FINLAND, Antti.Liski@tut.fi

## ABSTRACT

In model selection one attempts to use the data to find a single "winning" model, whereas with model averaging (MA) one seeks a smooth compromise across a set of competing models. Most existing MA methods are based on estimation of single model weights using some appropriate criterion. The problem of selecting the best subset or subsets of predictor variables is a common challenge for a regression analyst. The number of candidate models may become huge and any approach based on estimation of all single weights may become computationally infeasible. Our approach is to convert estimation of model weights into estimation of shrinkage factors with trivial computational burden. We define the class of shrinkage estimators in view of MA and show that the estimators can be constructed using penalized least squares (LS) estimation by putting appropriate restrictions on the penalty function. The relationship between shrinkage and parameter penalization provides tools to build up computationally efficient MA estimators which are easy to implement into practice.

## 1. THE MODEL

Our framework is the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim (\mathbf{0}, \sigma^2 \mathbf{I}_n), \quad (1)$$

where  $\mathbf{X}$  and  $\mathbf{Z}$  are  $n \times p$  and  $n \times m$  matrices of nonrandom regressors,  $(\mathbf{X}, \mathbf{Z})$  is assumed to be of full column-rank  $p + m < n$ ,  $\boldsymbol{\beta}$  and  $\boldsymbol{\gamma}$  are  $p \times 1$  and  $m \times 1$  vectors of unknown parameters. Our interest is in the effect of  $\mathbf{X}$  on  $\mathbf{y}$ , that is, we want to estimate  $\boldsymbol{\beta}$  while the role of  $\mathbf{Z}$  is to improve the estimation of  $\boldsymbol{\beta}$ .

We will work with the canonical form of the model (1), where  $z$ -variables are orthogonalized by writing the systematic part of the model (1) as

$$\begin{aligned} \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} &= \mathbf{X}\boldsymbol{\alpha} + \mathbf{M}\mathbf{Z}\boldsymbol{\gamma} \\ &= \mathbf{X}\boldsymbol{\alpha} + \mathbf{U}\boldsymbol{\theta}, \end{aligned} \quad (2)$$

where  $\boldsymbol{\alpha} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\boldsymbol{\gamma}$ ,

$$\mathbf{M} = \mathbf{I}_n - \mathbf{P} \quad \text{and} \quad \mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \quad (3)$$

are symmetric idempotent matrices. Since  $(\mathbf{M}\mathbf{Z})'\mathbf{M}\mathbf{Z} = \mathbf{Z}'\mathbf{M}\mathbf{Z}$  is positive definite [15], then there exists a nonsingular matrix  $\mathbf{C}$  such that [9]

$$\mathbf{C}'\mathbf{Z}'\mathbf{M}\mathbf{Z}\mathbf{C} = (\mathbf{M}\mathbf{Z}\mathbf{C})'(\mathbf{M}\mathbf{Z}\mathbf{C}) = \mathbf{U}'\mathbf{U} = \mathbf{I}_m. \quad (4)$$

In (4)  $\mathbf{U} = \mathbf{M}\mathbf{Z}\mathbf{C}$  denotes the matrix of orthogonal canonical auxiliary regressors. Introducing the canonical auxiliary parameters  $\boldsymbol{\theta} = \mathbf{C}^{-1}\boldsymbol{\gamma}$  we can write in (2)

$$\mathbf{M}\mathbf{Z}\boldsymbol{\gamma} = \mathbf{M}\mathbf{Z}\mathbf{C}\mathbf{C}^{-1}\boldsymbol{\gamma} = \mathbf{U}\boldsymbol{\theta}.$$

## 2. MODEL AVERAGING

A least squares MA estimator for  $\boldsymbol{\beta}$  takes the form

$$\begin{aligned} \tilde{\boldsymbol{\beta}} &= \sum_{i=0}^M \lambda_i \hat{\boldsymbol{\beta}}_i = \sum_{i=0}^M \lambda_i (\hat{\boldsymbol{\beta}}_0 - \mathbf{Q}\mathbf{W}_i \hat{\boldsymbol{\theta}}) \\ &= \hat{\boldsymbol{\beta}}_0 - \mathbf{Q}\mathbf{W}\hat{\boldsymbol{\theta}}, \end{aligned} \quad (5)$$

where  $\hat{\boldsymbol{\beta}}_0 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ ,  $\mathbf{W} = \sum_{i=0}^M \lambda_i \mathbf{W}_i$  and  $\mathbf{Q} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\mathbf{C}$ . The weights

$$\lambda_i = \lambda_i(\mathbf{M}\mathbf{y}) \geq 0, \quad i = 0, 1, \dots, M,$$

are assumed to depend on the least squares residuals  $\mathbf{M}\mathbf{y}$  and  $\sum_{i=0}^M \lambda_i = 1$ . Note especially that  $\hat{\boldsymbol{\theta}}$  is a function of  $\mathbf{M}\mathbf{y}$ . The selection matrices  $\mathbf{W}_i$ ,  $0 \leq i \leq M$  are nonrandom  $m \times m$  diagonal matrices with diagonal elements 0 or 1 whereas  $\mathbf{W}$  is a random  $m \times m$  diagonal matrix with diagonal elements

$$\mathbf{w} = (w_1, \dots, w_m)', \quad 0 \leq w_i \leq 1, \quad i = 1, \dots, m.$$

The equivalence theorem of Danilov and Magnus [3] provides a useful representation for the expectation, variance and  $MSE$  of the estimator  $\tilde{\boldsymbol{\beta}}$  given in (5). The theorem was proved under the assumptions that the disturbances  $\varepsilon_1, \dots, \varepsilon_n$  are i.i.d.  $N(0, \sigma^2)$ . By the theorem

$$\begin{aligned} MSE(\tilde{\boldsymbol{\beta}}) &= E[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})'] \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} + \mathbf{Q}[MSE(\mathbf{W}\hat{\boldsymbol{\theta}})]\mathbf{Q}'. \end{aligned}$$

The quality of  $\tilde{\boldsymbol{\beta}}$  essentially depends on statistical properties of the shrinkage estimator  $\mathbf{W}\hat{\boldsymbol{\theta}}$  and hence the relatively simple estimator  $\mathbf{W}\hat{\boldsymbol{\theta}}$  of  $\boldsymbol{\theta}$  characterizes the important features of the more complicated estimator  $\tilde{\boldsymbol{\beta}}$  of  $\boldsymbol{\beta}$ . It can be shown (Hansen [8]) that a least squares MA estimator like (5) can achieve lower  $MSE$  than any individual LS estimator.

### 3. PENALIZED LS AND SHRINKAGE

We introduce a set  $\mathcal{S}$  of shrinkage estimators for  $\beta$  and characterize them by using penalized least squares technique. Then we derive the efficiency bound for the shrinkage estimators with respect to  $MSE$  (mean squared error) when observations follow the normal distribution. Our aim is to find estimators whose  $MSE$  is uniformly as close to the efficiency bound as possible. It turns out that many interesting known estimators, like for example the soft and firm thresholding estimators, non-negative garrote [2] and the SCAD (smoothly clipped absolute deviation, [6]) estimators belong to this shrinkage class  $\mathcal{S}$ . On the other hand, for example the hard thresholding rule (pre testing) and the ridge estimator do not belong to  $\mathcal{S}$ .

Fitting the orthogonalized model (2) can be considered as a two-step least squares procedure [15]. The first step is to calculate  $\hat{\beta}_0 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$  and to replace  $\mathbf{y}$  by  $\mathbf{y} - \mathbf{X}\hat{\beta}_0 = \mathbf{M}\mathbf{y}$ , where  $\mathbf{M}$  is defined in (3). Then denote  $\mathbf{z} = \mathbf{U}'\mathbf{y}$ , and note from the definition of  $\mathbf{U}$  in (4) the equality  $\mathbf{U}'\mathbf{M} = \mathbf{U}'$ . Then the model (2) takes the form

$$\mathbf{z} = \boldsymbol{\theta} + \mathbf{U}'\boldsymbol{\varepsilon}, \quad \mathbf{U}'\boldsymbol{\varepsilon} \sim (\mathbf{0}, \sigma^2\mathbf{I}_m). \quad (6)$$

The second step is to estimate  $\boldsymbol{\theta}$  from the model (6).

Magnus et al. [13] estimated the weights  $0 \leq w_i \leq 1$ ,  $i = 1, \dots, m$  in (5) using a Bayesian technique, and decided on to advocate the Laplace estimator which is of a shrinkage type. Such estimators are computationally superior to estimators that require estimation of every single model weight  $\lambda_i$  in (5). We are now ready to define the important class  $\mathcal{S}$  of shrinkage estimators for  $\theta$  which we call simply shrinkage estimators.

**Definition** A real valued estimator  $\delta$  of  $\theta$  is a shrinkage estimator if the following four conditions hold:

- (a)  $0 \leq \delta(\hat{\theta}) \leq \hat{\theta}$  for  $\hat{\theta} \geq 0$ ,
- (b)  $\delta(-\hat{\theta}) = -\delta(\hat{\theta})$ ,
- (c)  $\delta(\hat{\theta})/\hat{\theta}$  is nondecreasing on  $[0, \infty)$  and
- (d)  $\delta(\hat{\theta})$  is continuous,

where  $\hat{\theta}$  is the LS estimator of  $\theta$ .

In estimation of  $\boldsymbol{\theta}$  we will use the penalized LS technique. If the penalty function satisfies proper regularity conditions, the penalized LS yields a solution which is a shrinkage estimator of  $\boldsymbol{\theta}$ . In this approach we choose a suitable penalty function in order to get a shrinkage estimator with good risk properties. The penalized least squares estimate (PenLS) of  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)'$  is the minimizer of

$$\frac{1}{2} \sum_{i=1}^m (z_i - \theta_i)^2 + \sum_{i=1}^m p_\lambda(|\theta_i|), \quad (7)$$

where  $\lambda > 0$ . It is assumed that the penalty function  $p_\lambda(\cdot)$  is

- (i) nonnegative,
  - (ii) nondecreasing and
  - (iii) differentiable on  $[0, \infty)$ .
- (8)

Minimization of (7) is equivalent to minimization componentwise. Thus we may simply minimize

$$l(\theta) = \frac{1}{2}(z - \theta)^2 + p_\lambda(|\theta|) \quad (9)$$

with respect to  $\theta$ .

**Example** There are close connections between the PenLS and variable selection or the PenLS and ridge regression, for example. Taking the  $L_2$  penalty  $p_\lambda(|\theta|) = \frac{\lambda}{2}|\theta|^2$  yields the ridge estimator

$$\check{\theta}_R = \frac{1}{1 + \rho}z,$$

where  $\rho > 0$  depends on  $\lambda$ . The hard thresholding penalty function

$$p_\lambda(|\theta|) = \lambda^2 - \frac{1}{2}(|\theta| - \lambda)^2 \mathbf{I}(|\theta| < \lambda)$$

yields the hard thresholding rule

$$\check{\theta}_H = z \{\mathbf{I}(|z| > \lambda)\}, \quad (10)$$

where  $\mathbf{I}(\cdot)$  is the indicator function. Then the minimizer of the expression (7) is  $z_j \{\mathbf{I}(|z_j| > \lambda)\}$ ,  $j = 1, \dots, m$ , and it coincides with the best subset selection for orthonormal designs. In statistics (see e.g. Morris et al. [14]) and in econometrics (see, e.g. Judge *et al.* [10]), the hard thresholding rule is traditionally called the pretest estimator.

The following theorem gives sufficient conditions for the PenLS estimate  $\check{\theta}$  of  $\theta$  to be a shrinkage estimator. Further, the theorem provides the lower bound of the mean squared error

$$MSE(\theta, \check{\theta}) = E[\check{\theta}(z) - \theta]^2. \quad (11)$$

This lower bound is called the *efficiency bound*.

**Theorem 3.1.** *We assume that the penalty function  $p_\lambda(\cdot)$  satisfies the assumptions (8). We make two assertions.*

(i) *If the three conditions hold*

- (1) *the function  $-\theta - p'_\lambda(\theta)$  is strictly unimodal on  $[0, \infty)$ ,*
- (2)  *$p'_\lambda(\cdot)$  is continuous and nonincreasing on  $[0, \infty)$ , and*
- (3)  $\min_{\theta} \{\theta + p'_\lambda(|\theta|)\} = p'_\lambda(0)$ ,

*then the PenLS estimate  $\check{\theta}$  of  $\theta$  belongs to the shrinkage family  $\mathcal{S}$ .*

(ii) If the conditions of the assertion (i) hold and  $z$  follows the normal distribution  $N(0, \sigma^2)$ , where  $\sigma^2$  is known, the efficiency bound of  $\check{\theta}$  is

$$\inf_{\check{\theta} \in \mathcal{S}} MSE(\theta, \check{\theta}) = \frac{\theta^2}{1 + \theta^2}. \quad (12)$$

Note that the pretest estimator  $\check{\theta}_H$  given in (10) is not continuous, and hence it does not belong to the class of shrinkage estimators  $\mathcal{S}$ . Magnus [11] demonstrates a number of undesirable properties of the pretest estimator. It is inadmissible and there is a range of values for which the  $MSE$  of  $\check{\theta}_H$  is greater than the  $MSE$  of both the least squares estimator  $\hat{\theta}(z) = z$  and the null estimator  $\hat{\theta}(z) \equiv 0$ . The traditional pretest at the usual 5% level of significance results in an estimator that is close to having worst possible performance with respect to the  $MSE$  criterion in the neighborhood of the value  $|\theta/\sigma| = 1$  which was shown to be of crucial importance.

**Example** The  $L_q$  penalty  $p_\lambda(|\theta|) = \lambda |\theta|^q$ ,  $q \geq 0$  results in a bridge regression [7]. The derivative  $p'_\lambda(\cdot)$  of the  $L_q$  penalty is nonincreasing on  $[0, \infty)$  only when  $q \leq 1$  and the solution is continuous only when  $q \geq 1$ . Therefore, only  $L_1$  penalty in this family yields a shrinkage estimator. This estimator is the soft thresholding rule, proposed by Donoho and Johnstone [4],

$$\check{\theta}_S = \text{sgn}(z)(|z| - \lambda)_+, \quad (13)$$

where  $z_+$  is shorthand for  $\max\{z, 0\}$ . LASSO [16] is the PenLS estimate with the  $L_1$  penalty in the general least squares and likelihood settings.

If the PenLS estimators satisfy the conditions of Theorem 3.1, the efficiency bound is known and the *regret* of  $\check{\theta}(z)$  can be defined as

$$r(\theta, \check{\theta}) = MSE(\theta, \check{\theta}) - \frac{\theta^2}{1 + \theta^2}.$$

We wish to find an estimator with the desirable property that its  $MSE$  is uniformly close to the infeasible efficiency bound. In theoretical considerations  $\sigma^2$  is assumed to be known, and hence we can always consider the variable  $z/\sigma$ . Then the expectation  $E$  is simply taken with respect to the  $N(\theta, 1)$  distribution (cf. Figure 1), and comparison of estimators risk performance is done under this assumption. In practical applications we replace the unknown  $\sigma^2$  with  $s^2$ , the estimate of  $\sigma^2$  in the unrestricted model. Danilov [3] demonstrated that effects of estimating  $\sigma^2$  are small in case of Laplace estimator. We expect the approximation to be accurate for other shrinkage estimators too, although more work is needed to clarify this issue.

### 3.1. Good PenLS shrinkage estimators

In this subsection we consider properties of two well known PenLS estimators which are shrinkage estimators. Bruce and Gao [1] compared the hard and soft thresholding rules

and showed that the hard thresholding rule tends to have bigger variance than the soft thresholding rule whereas soft thresholding tends to have bigger bias. To remedy the drawbacks of hard and soft thresholding, Fan and Li [6] suggested using continuous differentiable penalty function defined by

$$p'_\lambda(|\theta|) = \lambda \{I(|\theta| \leq \lambda) + \frac{(a\lambda - |\theta|)_+}{(a-1)\lambda} I(|\theta| > \lambda)\} \quad (14)$$

for some  $a > 2$  and  $\lambda > 0$ . If the penalty function in (7) is constant, i.e.  $p'(|\theta|) = 0$ , then the PenLS takes the form  $\check{\theta}(z) \equiv z$  which is unbiased. Since the SCAD penalty  $p'_\lambda(\theta) = 0$  for  $\theta > a\lambda$ , the resulting solution (Fan and Li [6])

$$\check{\theta}_{scad}(z) = \begin{cases} \text{sgn}(z)(|z| - \lambda)_+, & \text{if } |z| \leq 2\lambda, \\ \frac{(a-1)z - \text{sgn}(z)a\lambda}{(a-2)}, & \text{if } 2\lambda < |z| \leq a\lambda, \\ z, & \text{if } |z| > a\lambda \end{cases} \quad (15)$$

tends to be unbiased for large values of  $z$ . The estimator (15) can be viewed as a combination of soft thresholding for "small"  $|z|$  and hard thresholding for "large"  $|z|$ , with a piecewise linear interpolation inbetween.

Breiman [2] applied the non-negative garrote rule

$$\check{\theta}_G(z) = \begin{cases} 0, & \text{if } |z| \leq \lambda, \\ z - \lambda^2/z, & \text{if } |z| > \lambda \end{cases} \quad (16)$$

to subset selection in regression to overcome the drawbacks of stepwise variable selection rule and ridge regression. It is straightforward to show that the soft thresholding (13), SCAD (15) and non-negative garrote (16) estimators belong to the shrinkage class  $\mathcal{S}$  (cf. Definition). The ordinary LS (OLS) estimator  $\hat{\theta}(z) \equiv z$  is a good candidate for large  $z$ , and hence we wish that for large  $z$  an estimator  $\check{\theta}(z)$  is close to  $z$  in the sense that  $z - \check{\theta}(z)$  converges to zero when  $|z|$  increases. It can be readily seen that the estimators  $\check{\theta}_{scad}$  and  $\check{\theta}_G$  have this property. For the soft thresholding rule  $z - \check{\theta}_S(z)$  converges to a positive constant, but not to zero.

### 3.2. The Laplace and Subbotin estimators

Magnus [12] addressed the question of finding an estimator of  $\theta$  which is admissible, has bounded risk, has good risk performance around  $\theta = 1$ , and is optimal or near optimal in terms of minimax regret when  $z \sim N(\theta, 1)$ . The Laplace estimator

$$\hat{\theta}_L(z) = z - h(y)c \quad (17)$$

proved to be such an estimator, when  $c = \log 2$  and  $h(\cdot)$  is a given antisymmetric monotonically increasing function on  $(-\infty, \infty)$  with  $h(0) = 0$  and  $h(\infty) = 1$ . The Laplace estimator is the mean of the posterior distribution of  $\theta|z$  when a Laplace prior for  $\theta$  with  $\text{median}(\theta) = 0$  and  $\text{median}(\theta^2) = 1$  is assumed. In search of a prior which appropriately reflects the notion of ignorance, Einmahl et al. [5] arrived at the Subbotin prior that belongs to the

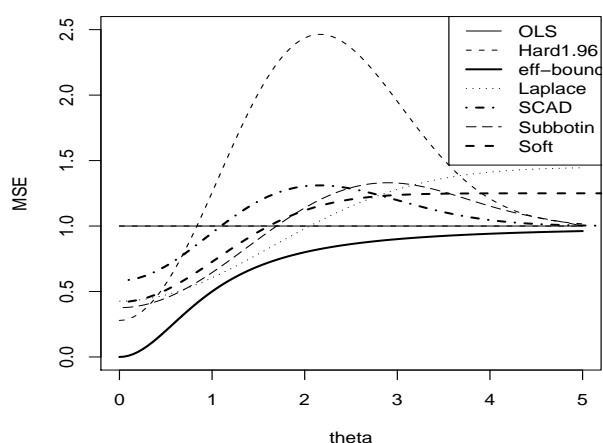


Figure 1.  $MSE$  of the OLS, the hard thresholding (10), Laplace (17), SCAD (15), Subbotin, soft thresholding estimators (13) and the efficiency bound (12) for the shrinkage estimators  $\mathcal{S}$ .

class of reflected gamma densities. In practical applications they recommended the Subbotin prior

$$\pi(\theta) = \frac{c^2}{4} e^{-c|\theta|^{1/2}}$$

with  $c = 1.6783$  which should stay close to the Laplace prior.

#### 4. CONCLUDING REMARKS

Many existing MA methods require estimation of every single model weight. For example, in regression analysis selection of the best subset from a set of  $m$  predictors, say, requires assessing  $2^m$  models, and consequently the computational burden soon increases too heavy when  $m$  becomes large.

It turns out, that the quality of the least squares MA estimator (5) depends on the shrinkage estimator of the auxiliary parameter  $\gamma$ . So, estimation of  $2^m$  model weights is converted into estimation of  $m$  shrinkage factors with trivial computational burden. We define the class of shrinkage estimators in view of MA and show that these shrinkage estimators can be constructed by putting appropriate restrictions on the penalty function. Utilizing the relationship between shrinkage and parameter penalization, we are able to build up computationally efficient MA estimators which are easy to implement into practice. These estimators include some well known estimators, like the non-negative garrote of Breiman [2], the lasso-type estimator of Tibshirani [16] and the SCAD estimator of Fan and Li [6]. In the simulation experiments we have assessed the quality of estimators in terms of estimated  $MSE$ 's. In this competition the winners were the SCAD and non-negative garrote but the Laplace estimator did almost as well. However, the results of the simulation study are not reported here.

#### 5. REFERENCES

- [1] Bruce, A. G. and Gao, H.-Y. (1996). Understanding WaveShrink: Variance and bias estimation. *Biometrika*, 83, 727–745.
- [2] Breiman, L. (1995). Better subset regression using nonnegative garrote. *Technometrics*, 37, 373–384.
- [3] Danilov, D. and Magnus, J. R. (2004). On the harm that ignoring pretesting can cause. *Journal of Econometrics*, 122, 2746.
- [4] Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81, 425–456.
- [5] Einmahl, J. H. J., Kumar, K. and Magnus J. R. (2011) Bayesian model averaging and the choice of prior. *CentER Discussion Paper*, No. 2011-003.
- [6] Fan, J. and Li, R. (2001). Variable Selection via Non-concave Penalized Likelihood and Its Oracle Properties. *Journal of the American statistical Association*, 96, 1348–1360.
- [7] Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35, 109–148.
- [8] Hansen, B. E. (2007). Least squares model averaging. *Econometrica*, 75, 1175–1189.
- [9] Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge, Cambridge University Press.
- [10] Judge, G. G., Griffiths, W. E., Hill, R. C., Lutkepohl, H. and Lee, T. C. (1985). *The Theory and Practice of Econometrics*, Wiley, New York.
- [11] Magnus, J. R. (1999). The traditional pretest estimator. *Theory of Probability and Its Applications*, 44, 293308.
- [12] Magnus, J. R. (2002). Estimation of the mean of a univariate normal distribution with a known variance. *Econometrics Journal*, 5, 225236.
- [13] Magnus, J. R., Powell, O. and Prüfer, P. (2010). A comparison of two model averaging techniques with an application to growth empirics. *Journal of Econometrics*, 154, 139–153.
- [14] Morris, C., Radhakrishnan, R. and Sclove, S. L. (1972). Nonoptimality of preliminary test estimators for the mean of a multivariate normals distribution. *Annals of Mathematical Statistics*, 43, 1481–1490.
- [15] Seber, G. A. F. (1977). *Linear Regression Analysis*, New York, Wiley.
- [16] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society B*, 1, 267–288.

# PUTTING BAYES TO SLEEP

Wouter M. Koolen<sup>1</sup> and Dimitri Adamskiy<sup>1</sup> and Manfred K. Warmuth<sup>2</sup>

<sup>1</sup> Computer Learning Research Centre and Department of Computer Science,  
Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK

<sup>2</sup> Department of Computer Science, University of California Santa Cruz, CA 95064, USA

## ABSTRACT

Consider sequential prediction algorithms that are given the predictions from a set of models as inputs. If the nature of the data is changing over time in that different models predict well on different segments of the data, then adaptivity is typically achieved by mixing into the weights in each round a bit of the initial prior (kind of like a weak restart). However, what if the favored models in each segment are from a *small subset*, i.e. the data is likely to be predicted well by models that predicted well before? Curiously, fitting such “sparse composite models” is achieved by mixing in a bit of all the past posteriors. This self-referential updating method is rather peculiar, but it is efficient and gives superior performance on many natural data sets. Also it is important because it introduces a long-term memory: any model that has done well in the past can be recovered quickly. While Bayesian interpretations can be found for mixing in a bit of the initial prior, no Bayesian interpretation is known for mixing in past posteriors.

We build atop the “specialist” framework from the on-line learning literature to give the Mixing Past Posteriors update a proper Bayesian foundation. We apply our method to a well-studied multitask learning problem and obtain a new intriguing efficient update that achieves a significantly better bound.

## 1. INTRODUCTION

We consider sequential prediction of outcomes  $y_1, y_2, \dots$  using a set of models  $m = 1, \dots, M$  for this task. In practice  $m$  could range over a mix of human experts, parametric models, or even complex machine learning algorithms. In any case we denote the prediction of model  $m$  for outcome  $y_t$  given past observations  $y_{<t} = (y_1, \dots, y_{t-1})$  by  $P(y_t|y_{<t}, m)$ . The goal is to design a computationally efficient predictor  $P(y_t|y_{<t})$  that maximally leverages the predictive power of these models as measured in log loss. The yardstick in this paper is a notion of *regret* defined w.r.t. a given *comparator class* of models or composite models: it is the additional loss of the predictor over the best comparator. For example if the comparator class is the set of base models  $m = 1, \dots, M$ , then the regret for a sequence of  $T$  outcomes  $y_{\leq T} = (y_1, \dots, y_T)$  is

$$\mathcal{R} := \sum_{t=1}^T -\ln P(y_t|y_{<t}) - \min_{m=1}^M \sum_{t=1}^T -\ln P(y_t|y_{<t}, m).$$

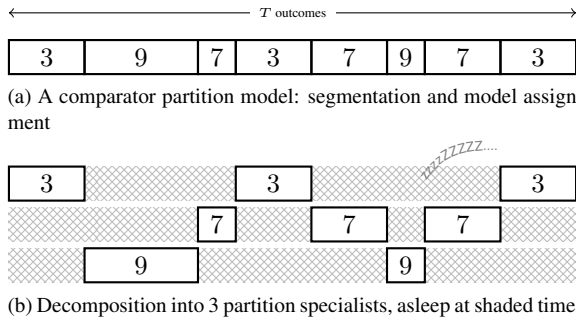
The Bayesian predictor with uniform model prior has regret at most  $\ln M$  for all  $T$ .

Now assume the nature of the data is changing with time: in an initial segment one model predicts well, followed by a second segment in which another model has small loss and so forth. For this scenario the natural comparator class is the set of *partition models* which divide the sequence of  $T$  outcomes into  $B$  segments and specify the model that predicts in each segment. By running Bayes on all exponentially many partition models comprising the comparator class, we can guarantee regret  $\ln \binom{T-1}{B-1} + B \ln M$ . The goal then is to find *efficient* algorithms with approximately the same guarantee as full Bayes. In this case this is achieved by the Fixed Share [1] predictor. It assigns a certain prior to all partition models for which the exponentially many posterior weights collapse to  $M$  posterior weights that can be maintained efficiently. Modifications of this algorithm achieve essentially the same bound for all  $T, B$  and  $M$  simultaneously [2, 3].

In an open problem Yoav Freund [4] asked whether there are algorithms that have small regret against *sparse* partition models where the base models allocated to the segments are from a small subset of  $N$  of the  $M$  models. The Bayes algorithm when run on all such partition models achieves regret  $\ln \binom{M}{N} + \ln \binom{T-1}{B-1} + B \ln N$ , but contrary to the non-sparse case, emulating this algorithm is NP-hard. However in a breakthrough paper, Bousquet and Warmuth in 2001 [4] gave the efficient MPP algorithm with only a slightly weaker regret bound. Like Fixed Share, MPP maintains  $M$  “posterior” weights, but it instead mixes in a bit of all past posteriors in each update. This causes weights of previously good models to “glow” a little bit, even if they perform bad locally. When the data later favors one of those good models, its weight is pulled up quickly. However the term “posterior” is a misnomer because no Bayesian interpretation for this curious self-referential update was known. Understanding the MPP update is a very important problem because in many practical applications [5, 6]<sup>1</sup> it significantly outperforms Fixed Share.

Our main philosophical contribution is finding a fully Bayesian interpretation for MPP. We employ the special-

<sup>1</sup>The experiments reported in [5] are based on precursors of MPP. However MPP outperforms these algorithms in later experiments we have done on natural data for the same problem (not shown).



ist framework from online learning [7, 8, 9]. So-called *specialist* models are either *awake* or *asleep*. When they are awake, they predict as usual. However when they are asleep, they “go with the rest”, i.e. they predict with the combined prediction of all awake models.

Instead of fully coordinated partition models, we construct *partition specialists* consisting of a base model and a set of segments where this base model is awake. The figure to the right shows how a comparator partition model is assembled from partition specialists. We can emulate Bayes on all partition specialists; the NP-completeness is avoided by forgoing a-priori segment synchronization. By carefully choosing the prior, the exponentially many posterior weights collapse to the small number of weights used by the efficient MPP algorithm. Our analysis technique magically aggregates the contribution of the  $N$  partition specialists that constitute the comparator partition, showing that we achieve regret close to the regret of Bayes when run on all full partition models. Actually our new insights into the nature of MPP result in slightly improved regret bounds.

We then apply our methods to the online multitask learning problem where a small subset of models from a big set solve a large number of tasks. Again simulating Bayes on all sparse assignments of models to tasks is NP-hard. We split an assignment into *subset specialists* that assign a single base model to a subset of tasks. With the right prior, Bayes on these subset specialists again gently collapses to an efficient algorithm with a regret bound not much larger than Bayes on all assignments. This considerably improves the previous regret bound of [10]. Our algorithm simply maintains one weight per model/task pair and does not rely on sampling (often used for multitask learning).

Why is this line of research important? We found a new intuitive Bayesian method to quickly recover information that was learned before, allowing us to exploit sparse composite models. Moreover, it expressly avoids computational hardness by splitting coordinated composite models into smaller constituent “specialists” that are asleep in time steps outside their jurisdiction. This method clearly beats Fixed Share when *few* base models constitute a partition, i.e. the composite models are sparse.

We expect this methodology to become a main tool for making Bayesian prediction adapt to sparse models. The goal is to develop general tools for adding this type of adaptivity to existing Bayesian models without losing

efficiency. It also lets us look again at the updates used in Nature in a new light, where species/genes cannot dare adapt too quickly to the current environment and must guard themselves against an environment that changes or fluctuates at a large scale. Surprisingly these type of updates might now be amenable to a Bayesian analysis. For example, it might be possible to interpret sex and the double stranded recessive/dominant gene device employed by Nature as a Bayesian update of genes that are either awake or asleep.

## 2. REFERENCES

- [1] Mark Herbster and Manfred K. Warmuth, “Tracking the best expert,” *Machine Learning*, vol. 32, pp. 151–178, 1998.
- [2] Paul A.J. Volf and Frans M.J. Willems, “Switching between two universal source coding algorithms,” in *Proceedings of the Data Compression Conference, Snowbird, Utah*, 1998, pp. 491–500.
- [3] Wouter M. Koolen and Steven de Rooij, “Combining expert advice efficiently,” in *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, Rocco Servedio and Tong Zang, Eds., June 2008, pp. 275–286.
- [4] Olivier Bousquet and Manfred K. Warmuth, “Tracking a small set of experts by mixing past posteriors,” *Journal of Machine Learning Research*, vol. 3, pp. 363–396, 2002.
- [5] David P. Helmbold, Darrell D. E. Long, Tracey L. Sconyers, and Bruce Sherrod, “Adaptive disk spin-down for mobile computers,” *ACM/Baltzer Mobile Networks and Applications (MONET)*, pp. 285–297, 2000.
- [6] Robert B. Gramacy, Manfred K. Warmuth, Scott A. Brandt, and Ismail Ari, “Adaptive caching by refetching,” in *NIPS*, Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, Eds. 2002, pp. 1465–1472, MIT Press.
- [7] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth, “Using and combining predictors that specialize,” in *Proc. 29th Annual ACM Symposium on Theory of Computing*. 1997, pp. 334–343, ACM.
- [8] Alexey Chernov and Vladimir Vovk, “Prediction with expert evaluators’ advice,” in *Proceedings of the 20th international conference on Algorithmic learning theory*, Berlin, Heidelberg, 2009, ALT’09, pp. 8–22, Springer-Verlag.
- [9] Alexey Chernov, Yuri Kalnishkan, Fedor Zhdanov, and Vladimir Vovk, “Supermartingales in prediction with expert advice,” *Theor. Comput. Sci.*, vol. 411, no. 29–30, pp. 2647–2669, June 2010.

- [10] Jacob Duan Abernethy, Peter Bartlett, and Alexander Rakhlin, “Multitask learning with expert advice,” Tech. Rep., University of California at Berkeley, Jan. 2007.

# ROBUST MODEL SELECTION FOR STOCHASTIC PROCESSES

*J. E. García<sup>1a</sup> V. A. González-López<sup>1b</sup> and M. L. L. Viola<sup>2</sup>*

<sup>1</sup>University of Campinas, Brazil.

<sup>a</sup>jg@ime.unicamp.br, <sup>b</sup>veronica@ime.unicamp.br,

<sup>2</sup>Federal University of São Carlos, Brazil.

## ABSTRACT

In this paper we address the problem of model selection for the set of finite memory stochastic processes with finite alphabet, when the data is contaminated. We consider  $m$  independent samples, with most of them being realizations of the same stochastic process with law  $Q$ , which is the one we want to retrieve. We devise a model selection procedure such that for a sample size large enough, the selected process is the one with law  $Q$ . Our model selection strategy is based on estimating relative entropies to select a subset of samples that are realizations of the same law. Although the procedure is valid for any family of finite order Markov models, we will focus on the family of variable length Markov chain models, which include the fixed order Markov chain model family. We define the asymptotic breakdown point  $\gamma$  for a model selection procedure, and we show the value  $\gamma$  for our procedure. This means that if the proportion of contaminated samples is smaller than  $\gamma$ , then, as the sample size grows our procedure selects a model for the process with law  $Q$ .

## 1. INTRODUCTION

In this paper we propose a robust strategy to select models from samples coming from a process which is contaminated and it is a discrete time stochastic process, on a finite alphabet. We will only consider the family of variable length Markov chain models, from now on VLMC (see [4, 1, 2, 5]) because it includes the fixed order Markov chain models and the independent case. For VLMC model selection we will use the version of the CTM algorithm introduced by [2], which is based on the Bayesian Information Criterion (BIC). It has been shown by [3] that a small Bernoulli random perturbation on a sample produced by a VLMC will effectively transform the process to an infinity memory process. They also show a variation of the original context algorithm given by [4] which can recover the VLMC model of the original chain, provided that the noise is small enough.

In this work we consider a different kind of contamination, we have a set of  $m$  independent samples, with most of them being from the same stochastic process with law  $Q$ , whose model we want to recover. The approach of this paper can be applied yet in the case in which we have only one sample produced by the concatenation of realizations of a mixture process which is the process  $Q$  plus a con-

taminant process. We define the asymptotic breakdown point  $\gamma$  for the model selection problem and we show the value of  $\gamma$  for our procedure.

Our procedure can be applied when the data is coming from a mixture of stochastic processes, for example in the problem of classification of languages according to their rhythmic features, using speech samples. The usual procedure to deal with this topic has been choose a subset of the original sample which seems best represent each language. Instead, if we apply this kind of robust procedure can be taken the complete dataset, see [6].

## 2. PRELIMINARIES

Let  $(X_t)$  be a discrete time stochastic process on a finite alphabet  $A$  with cardinal  $|A|$ . Denote the string (concatenation of elements from  $A$ )  $a_k a_{k+1} \dots a_r$  by  $a_k^r$ , where  $a_i \in A$ ,  $k \leq i \leq r$ . If the stochastic process  $(X_t)$  has probability law  $Q$ , and if  $x_1^n$  is a  $n$  realization of that process, we denote  $Q(x_1^n) = \text{Prob}(X_1^n = x_1^n)$ . The transition probability from the sequence  $x_1^n$  to the symbol  $a \in A$  is  $Q(a|x_1^n) = \text{Prob}(X_{n+1} = a | X_1^n = x_1^n)$ . Given a string  $s = a_k a_{k+1} \dots a_r$ , we denote its length as  $l(s) = r - k + 1$ . The empty string is denoted by  $\emptyset$  and  $l(\emptyset) = 0$ . We say that the string  $v$  is a postfix of a string  $s$  when there exists a string  $u$  such that  $s = uv$ . When  $s \neq v$ ,  $v$  is a proper postfix of  $s$ .

**Definition 1** A set  $\mathcal{T}$  of strings is called a tree if satisfies the following rules

1. no  $s_1 \in \mathcal{T}$  is a postfix of any other  $s_2 \in \mathcal{T}$ ,
2. no  $s_1 \in \mathcal{T}$  can be replaced by a proper postfix without violating rule 1.

We denote by  $d(\mathcal{T}) = \max(l(s), s \in \mathcal{T})$  the depth of the tree  $\mathcal{T}$ .

**Definition 2** Let  $(X_t)$  be a finite order stationary ergodic stochastic process on a finite alphabet  $A$  with probability law  $Q$ . We will say that the tree  $\mathcal{T}$  is a context tree for  $(X_t)$  if for any  $n \geq d(\mathcal{T})$  and for any sequence of symbols in  $A$ ,  $x_1^n$ , there exist a postfix  $s \in \mathcal{T}$  such that

$$Q(a|x_1^n) = Q(a|s), \quad \forall a \in A, \quad (1)$$

and no proper postfix of  $s$  satisfies equation (1). In that case  $s$  is called a context for the process  $Q$ .



**Definition 3** We will say that the stochastic process  $(X_t)$  is a variable length Markov chain compatible with the context tree  $\mathcal{T}$  if it verify definition 2.

Each model in the family of variable length Markov chain models, is identified by its context tree. For more details see [4, 1]. There are diverse methodologies for the selection and estimation of context trees, see for example [1, 2, 4, 5]. The context tree maximization CTM algorithm proposed by [2] is based on the BIC criterion and it will be used in this work for the statistical estimation of context trees.

For a given value  $D$  with  $n > D$ , if  $s$  is some string  $l(s) < D$ ,  $a \in A$  we denote by  $N_n(s, a)$  the number of occurrences of the string  $s$  followed by  $a$  in the sample  $x_1^n$ ,  $N_n(s, a) = |\{i : D < i \leq n, x_{i-D}^{i-1} = s, x_i = a\}|$ . The number of occurrences of  $s$  in the sample  $x_1^n$  is denoted by  $N_n(s)$  and  $N_n(s) = |\{i : D < i \leq n, x_{i-D}^{i-1} = s\}|$ . We denote by  $\mathcal{K}(x_1^n, D)$  the family of feasible context trees, where a feasible context tree  $\mathcal{T}$  is such that  $d(\mathcal{T}) \leq D$  and  $N_n(s) \geq 1$  for all  $s \in \mathcal{T}$  and for each string  $s'$  with  $N_n(s') \geq 1$  it has a postfix  $s \in \mathcal{T}$ . Now we can define the context tree estimator

$$\hat{\mathcal{T}}(x_1^n) = \arg \max_{\mathcal{T} \in \mathcal{K}(x_1^n, D)} \prod_{s \in \mathcal{T}} \tilde{P}_s(x_1^n) \quad (2)$$

where  $\tilde{P}_s(x_1^n) = n^{-\binom{|A|-1}{2}} \tilde{P}_{\text{ML},s}(x_1^n)$ .  $\tilde{P}_{\text{ML},s}(x_1^n) = \prod_{a \in A} \left( \frac{N_n(s,a)}{N_n(s)} \right)^{N_n(s,a)}$  if  $N_n(s) \geq 1$  and  $\tilde{P}_{\text{ML},s}(x_1^n) = 1$  if  $N_n(s) = 0$ .

For fixed  $n$  is considered  $D = D(n) = \log(n)$ . For a finite memory Markov process,  $\hat{\mathcal{T}}(x_1^n)$  converges eventually almost surely to the true  $\mathcal{T}$  of the law  $Q$ . The algorithm in [2] allows to compute these estimators in  $O(n)$  time, and to compute them on-line for all  $i \leq n$  in  $o(n \log(n))$  time. According to the corollary 2.12 in [2] the empirical probabilities  $\hat{Q}_{\hat{\mathcal{T}}}(a|s) = \frac{N_n(s,a)}{N_n(s)}$ ,  $a \in A$ ,  $s \in \hat{\mathcal{T}}$  converges to the true conditional probabilities  $Q(a|s)$ ,  $a \in A$ ,  $s \in \mathcal{T}$  almost surely as  $n \rightarrow \infty$ .

In order to simplify the notation we avoid the reference to the context tree  $\mathcal{T}$  (or  $\hat{\mathcal{T}}$ ) when the underlying context tree is understood and we adopt the notation

$$\hat{Q} = \widehat{CTM}((x_t)_{t=1}^n)$$

to emphasize that the estimation uses the CTM algorithm.

### 3. RELATIVE ENTROPY

**Definition 4** Given two probability mass functions  $P(\cdot)$  and  $Q(\cdot)$ , the relative entropy is

$$D(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \left( \frac{P(x)}{Q(x)} \right).$$

**Remark 1** Let  $P(\cdot)$ ,  $Q(\cdot)$  be two probability functions. Then,  $D(P||Q) \geq 0$ . The equality occurs if and only if  $P(x) = Q(x)$ ,  $\forall x \in \mathcal{X}$ .

**Definition 5** Let  $\mathcal{T}_P$  and  $\mathcal{T}_Q$  be two context trees following the definition 2 with probability law  $P$  and  $Q$  respectively.  $\mathcal{T}_{PQ}$  is defined by all the strings from  $\mathcal{T}_P$  and  $\mathcal{T}_Q$ , such that  $\mathcal{T}_{PQ}$  satisfy the definition 1.

From the previous definition,

$\mathcal{T}_{PQ} = \{s \in \mathcal{T}_P \cup \mathcal{T}_Q : \bar{A}s' \in \mathcal{T}_P \cup \mathcal{T}_Q \text{ postfix of } s\}$ . From Theorem 3 (see [6]), using  $\mathcal{T}_{PQ}$  it is possible to express the entropy between two processes through its conditional entropies as  $D(P||Q) = \sum_{s \in \mathcal{T}_{PQ}} P(s) D(P(\cdot|s)||Q(\cdot|s))$ .

**Remark 2** For  $s \in \mathcal{T}_{PQ}$ , we observe that  $P(\cdot|s)$  is the usual probability when  $s \in \mathcal{T}_P$ . If  $s \notin \mathcal{T}_P$ ,  $\exists s_1 \in \mathcal{T}_P$  and  $x$  some string, such that  $s = xs_1$  and  $P(\cdot|s) = P(\cdot|s_1)$ .

### 4. ASYMPTOTIC BREAKDOWN POINT

**Assumption 1** For a family  $\mathcal{F}$  of stochastic processes, consider a collection  $\{(X_{i,t}), i = 1, \dots, m\}$  of  $m$  independent finite memory stationary processes belonging to  $\mathcal{F}$ , where  $(X_{i,t})$  has probability law  $Q_i$ . If  $\mathcal{J}_{Q_i} = \{j \in \{1, \dots, m\} : (X_{jt}) \sim Q_i\}$ , suppose that exists  $i_0$  such that  $\forall i \neq i_0$ ,  $|\mathcal{J}_{Q_{i_0}}| > |\mathcal{J}_{Q_i}|$ , with  $i, i_0 \in \{1, \dots, m\}$ ,  $Q_{i_0}$  will be called as majority law of  $\mathcal{F}$ . Denote by  $\mathcal{C}_n^m = \{(x_{1,t})_{t=1}^n, (x_{2,t})_{t=1}^n, \dots, (x_{m,t})_{t=1}^n\}$  a collection of  $m$  samples of size  $n$  from  $(X_{i,t}), i = 1, \dots, m$ .

Let  $\mathcal{S}$  be a strategy of estimation, i.e.

$$\mathcal{S} : \Omega_{\mathcal{F}} \rightarrow \mathcal{F}$$

where  $\Omega_{\mathcal{F}}$  is the sample space of processes in  $\mathcal{F}$  and  $\mathcal{S}(\mathcal{C}_n^m)$  denotes the value of the estimator from the sample collection  $\mathcal{C}_n^m$ .

**Remark 3** Under the Assumption 1,  $\mathcal{S}(\mathcal{C}_n^m)$  indicates some strategy to select a sample (from  $\mathcal{C}_n^m$ ) or some set of samples (the best ones) to make the estimation of the majority law  $Q_{i_0}$ .

We define now the asymptotic breakdown point of the model estimator  $\mathcal{S}(\mathcal{C}_n^m)$ .

**Definition 6** Under the Assumption 1, the model estimator  $\mathcal{S}(\mathcal{C}_n^m)$  has an asymptotic breakdown point equal to  $\gamma$  for the family  $\mathcal{F}$ , if  $\gamma$  is the smallest value into  $(0, 1]$  such that, if  $\frac{|\mathcal{J}_{Q_{i_0}}|}{m} < \gamma$  then,

$$\lim_{n \rightarrow \infty} \mathcal{S}(\mathcal{C}_n^m) \neq Q_{i_0}, \text{ almost surely.}$$

### 5. ESTIMATORS

Given the collection of samples  $\mathcal{C}_n^m$ , for each  $i \in \{1, \dots, m\}$  denote  $\hat{Q}_i = \widehat{CTM}((x_{i,t})_{t=1}^n)$  the model estimated from the sample  $(x_{i,t})_{t=1}^n$  using the algorithm introduced in [2]. For each  $i, j \in \{1, \dots, m\}$ , denote by  $\hat{d}_{(i|j)}(\mathcal{C}_n^m)$  the relative entropy between  $\hat{Q}_i$  and  $\hat{Q}_j$ , i.e.  $\hat{d}_{(i|j)}(\mathcal{C}_n^m) = D(\hat{Q}_i || \hat{Q}_j)$ . Define then,

$$\bar{d}_{(i,j)}(\mathcal{C}_n^m) = \frac{\hat{d}_{(i|j)}(\mathcal{C}_n^m) + \hat{d}_{(j|i)}(\mathcal{C}_n^m)}{2}$$

and

$$\hat{V}_j(\mathcal{C}_n^m) = \frac{1}{m} \sum_{i=1}^m \bar{d}_{(j,i)}(\mathcal{C}_n^m).$$

We will refer to  $\bar{d}_{(i,j)}(\mathcal{C}_n^m)$  as being the Symmetrized Relative Entropy (SRE) between the samples  $i$  and  $j$  from  $\mathcal{C}_n^m$ . We will also say that  $\hat{V}_j(\mathcal{C}_n^m)$  is the mean SRE between the sample  $j$  and the other samples in  $\mathcal{C}_n^m$ .

Now, sort in increasing order the set  $\{\hat{V}_j(\mathcal{C}_n^m), j = 1, \dots, m\}$  and call  $j_i^*(\mathcal{C}_n^m)$  the index of the sample in the  $i$ th position on the ordered set, i.e.

$$j_1^*(\mathcal{C}_n^m) = \arg \min_{j=1, \dots, m} \left\{ \hat{V}_j(\mathcal{C}_n^m) \right\},$$

also

$$j_m^*(\mathcal{C}_n^m) = \arg \max_{j=1, \dots, m} \left\{ \hat{V}_j(\mathcal{C}_n^m) \right\}.$$

**Remark 4** To evaluate  $D(\hat{Q}_i || \hat{Q}_j)$  it is used Theorem 3 (see [6]), replacing the true probabilities by its empirical estimators and taking by the set of strings, the common tree given by definition 5 using the estimated trees from  $\hat{Q}_i$  and  $\hat{Q}_j$ .

**Theorem 1** Under the Assumption 1, if the estimator  $\mathcal{S}(\mathcal{C}_n^m)$  is defined as being  $\hat{Q}_{j_i^*(\mathcal{C}_n^m)}$  for some natural number  $i < m/2$ , then,  $\mathcal{S}(\mathcal{C}_n^m)$  has asymptotic breakdown point equal to  $\frac{1}{2}$ .

(See details of the proof in [6]).

In terms of quality of estimation, we can use Theorem 1 in order to propose a better strategy that can take advantage of the best samples detected by  $\{\hat{V}_j(\mathcal{C}_n^m)\}$  to construct a more powerful estimator for the majority law  $Q_{i_0}$ .

**Definition 7** Under the Assumption 1, we define the  $\alpha$ -trimmed CTM model estimator for  $Q$  as being

$$\hat{Q}^\alpha = CTM \left( (x_{j_i^*(\mathcal{C}_n^m), t})_{t=1}^n, i = 1, \dots, [(1-\alpha)m] \right),$$

for  $\alpha$  such that  $[(1-\alpha)m] \geq 1$ . Where  $[(1-\alpha)m]$  is the integer part of  $(1-\alpha)m$ .

**Remark 5**  $\hat{Q}^\alpha$  computes the CTM estimator assuming the selected samples as independent, this means that to compute the occurrences of each string  $s$  followed by  $a \in A$  will be necessary compute  $\hat{Q}(a|s) = \frac{N_n^\alpha(s,a)}{N_n^\alpha(s)}$  with  $N_n^\alpha(s) = \sum_{i=1}^{[(1-\alpha)m]} N_n^i(s)$  and  $N_n^\alpha(s,a) = \sum_{i=1}^{[(1-\alpha)m]} N_n^i(s,a)$  where  $N_n^i$  are the occurrences computed from the sample  $(x_{j_i^*(\mathcal{C}_n^m), t})_{t=1}^n$ . Where each string  $s$  comes from the set of feasible trees, with the same restriction as was assumed for equation(2).

**Theorem 2** Under the Assumption 1, for  $\alpha$  such that  $[(1-\alpha)m] \geq 1$ . The estimator  $\mathcal{S}(\mathcal{C}_n^m)$  defined by  $\hat{Q}^\alpha$  has

- (i) an asymptotic breakdown point equal to  $\alpha$ , when  $\alpha \in (0, \frac{1}{2})$  and
- (ii) an asymptotic breakdown point equal to  $\frac{1}{2}$  when  $\alpha \in [\frac{1}{2}, 1]$ .

(See details of the proof in [6]).

**Remark 6** If  $\alpha = (1 - \frac{1}{m})$  the estimator is given by  $\hat{Q}_{j_1^*(\mathcal{C}_n^m)}$ , i.e. the most representative empirical law, because the sample  $(x_{j_1^*(\mathcal{C}_n^m), t})_{t=1}^n$  would be considered the most representative in terms of the mean SRE.

## 6. CONCLUSION

In this paper we introduce a strategy of robust estimation to estimate the majority law from a collection of samples coming from VLMC processes. That strategy takes advantage from the convergence ‘‘almost surely’’ guaranteed by the CTM algorithm, but it is not restricted to this algorithm and can be applied using other algorithms of estimation. From a practical point of view, the strategy takes advantage also from the structure of trees (of VLMC), because the structure of tree allows to express the relative entropy between two processes in terms of the conditional probabilities. Using a very convenient structure of tree, that is a composition between the trees of the two processes (from [6]) the strategy can be formulated as a precise calculus between the empirical probability laws. The strategy achieves the best level of robustness, that is at most 50% of contamination. In addition, the strategy reveals how to improve the estimation, doing to grow the number of samples used for it, with the selection of the best samples to do the estimation.

## 7. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support for this research provided by CNPq’s projects 485999/2007-2 and 476501/2009-1 and USP project ‘‘Mathematics, computation, language and the brain’’.

## 8. REFERENCES

- [1] P. Buhlmann and A. Wyner, *Ann. Statist.* **27**, 480 (1999).
- [2] I. Csiszár and Z. Talata, *IEEE Trans. Inform. Theory*, **52**, 1007 (2006).
- [3] P. Collet, A. Galves and F. Leonardi, *Electronic Journal of Probability*. **13**, 1345 (2008).
- [4] J. Rissanen, *IEEE Trans. Inform. Theory*, **29**(5) 656 (1983).
- [5] A. Galves, C. Galves, J. Garcia, N. L. Garcia and F. Leonardi, *Annals of Applied Statistics*, **6**(1) 186 (2012).
- [6] J. E. García, V. A. González-López and M.L.L. Viola, *Robust model selection for finite memory stochastic processes*. Submitted.

# SUMMARISING EVENT SEQUENCES WITH SERIAL EPISODES

*Jilles Vreeken and Nikolaj Tatti*

Department of Mathematics and Computer Science,  
University of Antwerp, Belgium,  
{firstname.lastname}@ua.ac.be

## ABSTRACT

The discovery of patterns is an important aspect of data mining. Data mining is the field of research concerned with the extraction of useful insight from large databases. The process of finding patterns in data is called pattern mining. A pattern can be any type of regularity in the data, such as, e.g., items are typically sold together, or events that often happen in close vicinity. An ideal outcome of pattern mining is a small set of patterns, containing no redundancy or noise, that identifies the key structure of the data.

We pursue this ideal for sequential data, employing a *pattern set* mining approach. We employ the MDL principle to identify the best set of sequential patterns, and propose two approaches for mining good pattern sets: the first algorithm selects a good pattern set from a large candidate set, while the second is a parameter-free any-time algorithm that mines pattern sets directly from the data. Experimentation on synthetic and real data demonstrates we efficiently discover small sets of informative patterns.

## 1. INTRODUCTION

Suppose we have an event sequence database, and are interested in its most important patterns. Traditionally, we would apply frequent pattern mining, and mine all patterns that occur at least so-many times. For non-trivial thresholds, however, by the pattern explosion we would then be buried in huge amounts of highly redundant patterns—making the patterns the problem instead of the solution.

We therefore adopt a different approach. Instead of considering patterns individually, which is where the explosion stems from, we are after the *set of patterns* that summarises the data best. Desired properties of such a summary include that it should be small, generalise the data well, and be non-redundant. To this end, we employ the Minimum Description Length principle [1], by which we can identify the best set of patterns as the set by which we can describe the data most succinctly.

This approach has been shown to be highly successful for transaction data [2], where the discovered patterns provide insight, as well as high performance in a wide range of data mining tasks, including clustering, missing value estimation, and anomaly detection.

Sequence data, however, poses additional challenges over binary data. For starters, event orders are important, and we have to take gaps in patterns into account. As such,

encoding the data given a cover, finding a good cover given a set of patterns, as well as finding good sets of patterns, are all much more complicated for sequence data.

As we identify the best model by compression, and consider strings as data, standard compression approaches are related. However, although general purpose compressors provide top-notch compression, they do not result interpretable models. In our case, compression is not the goal, but a means for identifying those patterns that together describe the data most succinctly.

We here introduce a statistically well-founded approach for succinctly summarising event sequences, or SQS for short—pronounced as ‘squeeze’. We formalise how to encode a sequence dataset given a set of episodes, and formalise an MDL score for pattern sets. To optimise this score, we give an efficient heuristic to determine which pattern best describes what part of your data. To find good sets of patterns, we introduce two heuristics: SQS-CANDIDATES filters a given candidate collection, and SQS-SEARCH is a parameter-free any-time algorithm that efficiently mines models directly from data.

In this extended abstract we give a quick overview of SQS, only sketching the encoding and algorithms, and only report on some highlights of the empirical evaluation. For more detail, we refer the reader to [3].

## 2. MDL FOR EVENT SEQUENCES

As data type we consider *event sequences*. A sequence database  $D$  over an event alphabet  $\Omega$  consists of  $|D|$  sequences  $S \in D$ . Every  $S \in D$  is a sequence of  $|S|$  events  $e \in \Omega$ , i.e.  $S \in \Omega^{|S|}$ . We write  $S[i]$  to mean the  $i$ th event in  $S$  and  $S[i, j]$  to mean a subsequence  $S[i] \cdots S[j]$ . We denote by  $\|D\|$  the sum of the lengths of all  $S_i \in D$ , i.e.  $\|D\| = \sum_{S_i \in D} |S_i|$ . The support of an event  $e$  in  $S$  is its occurrences in  $S$ , i.e.  $\text{supp}(e | S) = |\{i \in S | i = e\}|$ , and the support of  $e$  in a database  $D$  is defined as  $\text{supp}(e | D) = \sum_{S \in D} \text{supp}(e | S)$ .

As patterns we consider serial episodes. A serial episode  $X$  is a sequence of events and we say that a sequence  $S$  contains  $X$  if there is a subsequence in  $S$  equal to  $X$ . Note that we are allowing gap events between the events of  $X$ . A singleton pattern is a single event  $e \in \Omega$ .

As models we consider *code tables*. A code table has four columns, one for patterns, one for pattern codes, and the latter two contain codes for indicating presence/absence

of a gap within a pattern. To ensure any sequence over  $\Omega$  can be encoded by a code table, we require that all the singleton events in the alphabet,  $X \in \Omega$ , are included in a code table  $CT$ .

### Encoding a Database

An encoded database consists of two code streams,  $C_p$  and  $C_g$ , that follow from the cover  $C$  chosen to encode the database. The first code stream, the pattern-stream, denoted by  $C_p$ , is a list of  $|C_p|$  codes,  $code_p(\cdot)$ , for patterns  $X \in CT$  corresponding to the patterns chosen by ‘cover’ algorithm. For example,  $code_p(a)code_p(b)code_p(c)$  encodes the sequence ‘abc’.

For  $L(code_p(X))$ , the lengths of pattern codes in  $C_p$ , as stored in the second column of  $CT$ , we use optimal prefix codes. Let us write  $usage(X)$  for how often  $code_p(X)$  occurs in  $C_p$ . That is,  $usage(X) = |\{Y \in C_p \mid Y = code_p(X)\}|$ . Then, the probability of  $code_p(X)$  in  $C_p$  is its relative occurrence in  $C_p$ . So, we have

$$L(code_p(X) \mid CT) = -\log \left( \frac{usage(X)}{\sum_{Y \in CT} usage(Y)} \right).$$

Serial episodes allow for gaps—only when we read the code for a singleton pattern  $X$  we can unambiguously append  $X$  to the decoded data. When  $X$  is a non-singleton pattern, we may only append the first symbol  $x_1$ , as before writing event  $x_2$  of  $X$ , we need to know whether or not one or more gap events occur in between.

This is what  $C_g$ , the gap code stream, encodes. It is a list of optimal prefix codes for gap occurrences/absences within pattern embeddings. These code lengths,  $L(code_g(X))$  and  $L(code_n(X))$ , are dependent on their relative frequency. Let us write  $gaps(X)$  to refer to the number of gap events within the usage of pattern  $X$  in the cover of  $D$ . We then resp. have  $fills(X) = usage(X)(|X| - 1)$ , for the number of non-gaps in the usage of pattern  $X$ , and

$$L(code_g(X) \mid CT) = -\log \left( \frac{gaps(X)}{gaps(X) + fills(X)} \right),$$

for the length of a gap code within a pattern  $X$ , and analogue for  $L(code_n(X) \mid CT)$ .

Combining the above, we straightforwardly arrive at  $L(C_p \mid CT) = \sum_{X \in CT} usage(X)L(code_p(X))$  for the encoded length of the pattern-stream, and analogously have

$$L(C_g \mid CT) = \sum_{\substack{X \in CT \\ |X| > 1}} \left( gaps(X)L(code_g(X)) + fills(X)L(code_n(X)) \right)$$

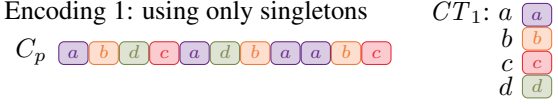
for  $C_g$ . We can then define  $L(D \mid CT)$ , the length of a database  $D$  given code table  $CT$  and cover  $C$  as

$$L(D \mid CT) = L_{\mathbb{N}}(|D|) + \sum_{S \in D} L_{\mathbb{N}}(|S|) + L(C_p \mid CT) + L(C_g \mid CT),$$

where  $|D|$  is the number of sequences in  $D$ , and  $|S|$  is the length of a sequence  $S \in D$ . To encode these values, we use  $L_{\mathbb{N}}$ , Rissanen’s universal code for integers [4].

Data  $D$ : a, b, d, c, a, d, b, a, a, b, c

Encoding 1: using only singletons



Encoding 2: using patterns

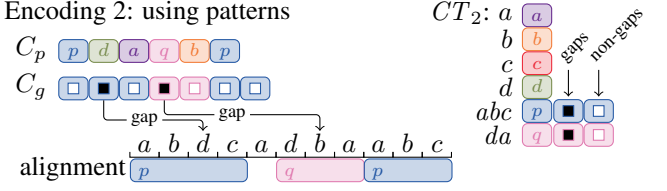


Figure 1. Toy example of two possible encodings. The first encoding uses only singletons. The second encoding uses singletons and two patterns, namely,  $abc$  and  $da$

*An Example.* Consider the toy example in Fig. 1. One possible encoding is to use only singletons, meaning that gap stream is empty. Another encoding is to use patterns. For example, to encode ‘abc’, we first give the code for  $abc$  in the pattern stream, then a no-gap code (white) in  $C_g$  to indicate  $b$ , then a gap code (black) in  $C_g$ , next the code for  $d$  in  $C_p$ , and we finish with a no-gap code in  $C_g$ .

### Encoding a Code Table

Next we discuss how to calculate  $L(CT)$ , the encoded length of a code table  $CT$ . We encode its number of entries using  $L_{\mathbb{N}}$ . For later use, and to avoid bias by large or small alphabets, we encode the number of singletons,  $|\Omega|$ , and the number of non-singleton entries,  $|CT \setminus \Omega|$ , separately. We disregard any non-singleton pattern with  $usage(X) = 0$ , as it is not used for describing the data.

The simplest valid code table consists of only singletons. We refer to this as the *standard code table*, or  $ST$ . We encode the patterns in the left-hand side column using  $ST$ , which allows us to decode up to the names of events.

The usage of  $Y \in ST$  is the support of  $Y$  in  $D$ . Hence, the code length of  $Y$  in  $ST$  is defined as  $L(code_p(Y) \mid ST) = -\log \frac{supp(Y|D)}{|D|}$ . Before we can use these codes, the recipient needs these supports. We transmit these by the index of a number composition, the number of combinations of summing to  $m$  with  $n$ , non-zero, terms. The length in bits of such an index is  $L_U(m, n) = \log \binom{m-1}{n-1}$ , where for  $m = 0$ , and  $n = 0$ , we define  $L_U(m, n) = 0$ .

We can now reconstruct the first column of  $CT$ . To encode a pattern  $X \in CT$ , the number of bits is the length of  $X$ ,  $|X|$ , and the sum of the singleton codes, i.e.  $L_{\mathbb{N}}(|X|) + \sum_{x_i \in X} L(code(x_i) \mid ST)$ .

Next, we encode the second column. To avoid bias, we treat the singletons and non-singleton entries of  $CT$  differently. Let us write  $\mathcal{P}$  to refer to the non-singleton patterns in  $CT$ , i.e.  $\mathcal{P} = CT \setminus \Omega$ . For the elements of  $\mathcal{P}$ , we first encode the sum of their usages, denoted by  $usage(\mathcal{P})$ , and use  $L_U$  identify the individual usages. Together with  $ST$ , we can reconstruct all usages in  $CT$ .

This leaves the gap-codes of  $CT$ , for which we encode  $gaps(X)$  using  $L_{\mathbb{N}}$ . The number of non-gaps then follows

from the length of a pattern  $X$  and its usage.

Together, we have  $L(CT | C, D)$ , the encoded size in bits of a code table  $CT$  for a cover  $C$  of a database  $D$ , as

$$\begin{aligned} L(CT | C) = & L_{\mathbb{N}}(|\Omega|) + L_U(|D|, |\Omega|) + \\ & L_{\mathbb{N}}(|\mathcal{P}| + 1) + L_{\mathbb{N}}(\text{usage}(\mathcal{P}) + 1) + \\ & L_U(\text{usage}(\mathcal{P}), |\mathcal{P}|) + \sum_{X \in \mathcal{P}} L(X, CT) \quad , \end{aligned}$$

where  $L(X, CT)$ , the encoded length for the events, length, and the number of gaps of a pattern  $X$  in  $CT$ , is

$$\begin{aligned} L(X, CT) \\ = & L_{\mathbb{N}}(|X|) + L_{\mathbb{N}}(\text{gaps}(X) + 1) + \sum_{x \in X} L(\text{code}_p(x | ST)) \end{aligned}$$

By MDL, we define the optimal set of serial episodes for a given sequence database as the set for which the optimal cover and associated optimal code table minimises

$$L(CT, D) = L(CT | C) + L(D | CT) .$$

More formally, we define the problem as follows.

**Minimal Code Table Problem** *Let  $\Omega$  be a set of events and let  $D$  be a sequence database over  $\Omega$ , find the minimal set of serial episodes  $\mathcal{P}$  such that for the optimal cover  $C$  of  $D$  using  $\mathcal{P}$  and  $\Omega$ , the total encoded cost  $L(CT, D)$  is minimal, where  $CT$  is the code-optimal code table for  $C$ .*

This problem entails a large search space. First of all, there are many different ways to cover a database given a set of patterns. Second, there are many sets of serial episodes  $\mathcal{P}$  we can consider. However, neither of these problems exhibits trivial structure that we can exploit for fast search, e.g. (weak) monotonicity.

### 3. COVERING A STRING

Encoding, or covering, a sequence is more difficult than decoding one. The reason is simple: when decoding there is no ambiguity, while when encoding there are many choices, i.e. what pattern to encode a symbol with. In other words, given a set of episodes, there are many valid ways to cover a sequence, where by our problem definition we are after the cover  $C$  that minimises  $L(CT, D)$ .

Assume we are decoding a sequence  $S_k \in D$ . Assume we decode the beginning of a pattern  $X$  at  $S_k[i]$  and that the last symbol belonging to this instance of  $X$  is, say,  $S_k[j]$ . We say that  $S_k[i, j]$  is an *active window* for  $X$ . Moreover, we can use FINDWINDOWS in [5] to discover all minimal windows for a pattern  $X$  in  $O(|X||D|)$ .

Let  $\mathcal{P}$  be the set of non-singleton patterns used by the encoding. We define an *alignment*  $A$  to be the set of all active windows for all non-singleton patterns  $X \in \mathcal{P}$ :  $A = \{(i, j, X, k) \mid S_k[i, j] \text{ is an active window for } X, S_k \in D\}$ . An alignment corresponding to the second encoding given in Figure 1 is  $\{(1, 4, abc, 1), (6, 8, da, 1), (9, 11, abc, 1)\}$ .

Note that an alignment  $A$  does not uniquely define the cover of the sequence, as it does not take into account how

the intermediate symbols (if any) within the active windows of a pattern  $X$  are encoded. However, an alignment  $A$  for a sequence database  $D$  does define an equivalence class over covers of the same encoded length. In fact, given a sequence database  $D$  and an alignment  $A$ , we can determine the number of bits our encoding scheme would require, as we can distill the  $\text{usage}(X)$  and  $\text{gaps}(X)$  from  $A$ . As such, given an alignment  $A$  for  $D$ , we can trivially construct a valid cover  $C$  for  $D$ , simply by following  $A$  and greedily covering  $S_k$  with pattern symbols if possible, and singletons otherwise. Likewise, we can derive the associated code-optimal code table  $CT$  for  $A$ .

In [3] we show that given a code table  $CT$ , we can find the alignment of  $D$  that minimises the encoded length using the code lengths in  $CT$ . With the above, we can then calculate the optimal codes for this new alignment. By iterating these steps, we can heuristically approximate the optimal cover of  $D$  given a set of patterns  $\mathcal{P}$ .

## 4. MINING CODE TABLES

With the above, we can score the quality of a pattern set, and heuristically optimise the alignment of a pattern set. This leaves us with the problem of finding good sets of patterns. We sketch our two algorithms to do so.

### 4.1. Filtering Candidates

Our first algorithm, SQS-CANDIDATES, assumes that we have a (large) set of candidate patterns  $\mathcal{F}$ . In practice, we assume the user obtains this set of patterns using a frequent pattern miner, although any set of patterns over  $\Omega$  will do. From  $\mathcal{F}$  we select that subset  $\mathcal{P} \subseteq \mathcal{F}$  such that the optimal alignment  $A$  and associated code table  $CT$  minimises  $L(D, CT)$ .

We sort candidates  $\mathcal{F}$  ascending by  $L(D, \{X\})$ . We then iteratively greedily test each pattern  $X \in \mathcal{F}$ . If adding  $X$  to  $\mathcal{P}$  improves the score, we keep  $X$  in  $\mathcal{P}$ , otherwise it is permanently removed.

Over time, new patterns can take over the role of older patterns. To this end, we prune redundant patterns after each successful addition. During pruning, we iteratively consider each pattern  $Y \in \mathcal{P}$  in order of insertion. If  $\mathcal{P} \setminus X$  improves the total encoded size, we remove  $X$  from  $\mathcal{P}$ . As testing every pattern in  $\mathcal{P}$  at every successful addition may become rather time-consuming, we use a simple heuristic: if the total gain of the windows of  $X$  is higher than the cost of  $X$  in the code table we do not test  $X$ .

After SQS-CANDIDATES considered every pattern of  $\mathcal{F}$ , we run one final round of pruning without this heuristic. Finally, we order the patterns in  $\mathcal{P}$  by  $L(D, \mathcal{P}) - L(D, \mathcal{P} \setminus X)$ . That is, by the impact on the total encoded length when removing  $X$  from  $\mathcal{P}$ . This order tells us which patterns in  $\mathcal{P}$  are most important.

### 4.2. Directly Mining Good Code Tables

The SQS-CANDIDATES algorithm requires a collection of candidate patterns to be materialised, which in practice can be troublesome; the well-known pattern explosion may prevent patterns to be mined at as low thresholds as desired.

We therefore propose an alternative strategy, that discovers good code tables directly from data. Instead of filtering a pre-mined candidate set, we now discover candidates on the fly, considering only patterns that we expect to optimise the score given the current alignment.

To illustrate the general idea, consider that we have a current set of patterns  $\mathcal{P}$ . We iteratively find patterns of form  $XY$ , where  $X, Y \in \mathcal{P} \cup \Omega$  producing the lowest  $L(D, \mathcal{P} \cup \{XY\})$ . We add  $XY$  to  $\mathcal{P}$  and continue until no gain is possible. Unfortunately, as testing each combination takes  $O((|\mathcal{P}| + |\Omega|)^2(|\mathcal{P}| + 1) \|D\|)$  time, we cannot do this exhaustively and exactly within reasonable time.

To guarantee the fast discovery of good candidates, we design a heuristic that, given a pattern  $P$ , will find a pattern  $PQ$  of high expected gain in only  $O(|\mathcal{P}| + |\Omega| + \|D\|)$ .

In [3], we show that if we take  $N$  active windows of  $P$ , and  $N$  active windows of  $Q$ , and convert them into  $N$  active windows of  $PQ$ , the difference in total encoded length can be calculated in constant time—as we know which  $N$  active windows to use: those with shortest length.

This gives the outline of SQS-SEARCH. We enumerate minimal windows of  $PQ$  from shortest to largest. At each step we compute the score using Proposition 3 of [3], and among these scores we pick optimal one. We can do this in linear time by considering the active windows of  $P$  ascending on length, ignoring all singleton gap elements, and counting all elements occurring right after  $P$ .

To save on computation, we do not iteratively consider the estimated optimal  $PQ$ , but instead iteratively compute and rank all  $PQ$  on estimated gain, consider these in turn, and recompute once the candidate pool is depleted. Like for SQS-CANDIDATES, we apply pruning after each accepted candidate, as well as at the end of the search.

## 5. EXPERIMENTS

We here give a quick taste of the results obtained with SQS, and refer the interested reader to the full publication for further empirical evaluation [3].

*Synthetic Data.* First, we consider the synthetic *Indep*, *P10*, and *P50* datasets. Each consists of a single sequence of 10 000 events over an alphabet of 1 000. In the former, all events are independent, whereas in the latter two we planted resp. 10 and 50 patterns of 5 events 10 times each, with 10% probability of having a gap between consecutive events, but are independent otherwise.

For the *Indep* dataset, though 9 000+ episodes occur at least twice, both methods correctly identify it does not contain significant structure. For *P10* both methods return the 10 patterns. *P50* has a very high density of pattern symbols (25%). SQS-CANDIDATES and SQS-SEARCH resp. find 47 and 46 patterns exactly, plus fragments, due to partial overwrites during generation, of the others.

*Real Data.* In order to interpret the patterns, we consider 788 abstracts of papers from the Journal of Machine Learning Research website. The events are the stemmed words from the text, with stop words removed. We obtain compression of about 30 000 bits, with 563 and 580 patterns respectively, more than two orders of magnitude less

Table 1. JMLR data. Top-10 patterns by SQS-SEARCH

patterns	$\Delta L$	patterns	$\Delta L$
1. supp. vec. mach.	850	6. large scale	329
2. machine learn.	646	7. near. neighbor	322
3. state [of the] art	480	8. dec. tree	293
4. data set	446	9. neural netw.	289
5. Bayesian netw.	374	10. cross val.	279

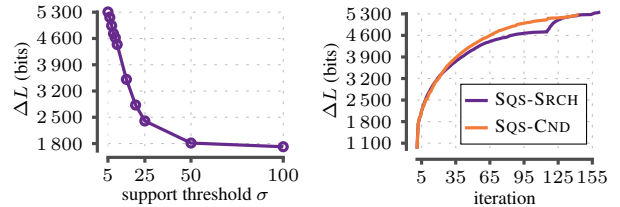


Figure 2. *Addresses* dataset,  $\Delta L$ . (left) varying support thresholds for SQS-CANDIDATES. (right) SQS-CANDIDATES and SQS-SEARCH per accepted candidate.

than the number of candidates for SQS-CANDIDATES.

Table 1 depicts the top-10 patterns most aiding compression, as found by SQS-SEARCH.  $\Delta L$  is the increase in bits the pattern would be removed from  $CT$ . The left-hand plot of Fig. 2, for SQS-CANDIDATES, shows the gain in compression for different support thresholds. Lower thresholds, i.e. richer candidate sets, allow for better models. In the right-hand plot, we compare SQS-CANDIDATES and SQS-SEARCH, showing the gain in bits over  $ST$  per candidate accepted into  $CT$ . Both search processes consider patterns aiding compression strongly first. The slight dip of SQS-SEARCH is by its batch-wise search.

## 6. CONCLUSION

Altogether, the long and the short of it is that SQS mines small sets of highly informative, non-redundant, serial episodes that succinctly describe the data at hand.

## 7. REFERENCES

- [1] Jorma Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 1, pp. 465–471, 1978.
- [2] Jilles Vreeken, Matthijs van Leeuwen, and Arno Siebes, “KRIMP: Mining itemsets that compress,” *Data Min. Knowl. Disc.*, vol. 23, no. 1, pp. 169–214, 2011.
- [3] Nikolaj Tatti and Jilles Vreeken, “The long and the short of it: Summarising event sequences with serial episodes,” in *KDD*, 2012.
- [4] Jorma Rissanen, “Modeling by shortest data description,” *Annals Stat.*, vol. 11, no. 2, pp. 416–431, 1983.
- [5] Nikolaj Tatti and Boris Cule, “Mining closed strict episodes,” *Data Min. Knowl. Disc.*, 2011.



# THE OPTIMALITY OF JEFFREYS PRIOR FOR ONLINE DENSITY ESTIMATION AND THE ASYMPTOTIC NORMALITY OF MAXIMUM LIKELIHOOD ESTIMATORS

Fares Hedayati<sup>1</sup> and Peter L. Bartlett<sup>2</sup>

<sup>1</sup> University of California at Berkeley,  
fareshed@eecs.berkeley.edu

<sup>2</sup> University of California at Berkeley,  
Queensland University of Technology, bartlett@cs.berkeley.edu

## ABSTRACT

We study online learning under logarithmic loss with regular parametric models. We show that a Bayesian strategy predicts optimally only if it uses Jeffreys prior. This result was known for canonical exponential families; we extend it to parametric models for which the maximum likelihood estimator is asymptotically normal. The optimal prediction strategy, normalized maximum likelihood, depends on the number  $n$  of rounds of the game, in general. However, when a Bayesian strategy is optimal, normalized maximum likelihood becomes independent of  $n$ . Our proof uses this to exploit the asymptotics of normalized maximum likelihood. The asymptotic normality of the maximum likelihood estimator is responsible for the necessity of Jeffreys prior.

## 1. INTRODUCTION

In the online learning setup, the goal is to predict a sequence of outcomes, revealed one at a time, almost as well as a set of experts. We consider online density estimators with log loss, where the forecaster's prediction at each round takes the form of a probability distribution over the next outcome, and the loss suffered is the negative logarithm of the forecaster's probability of the outcome. The aim is to minimize the regret, which is the difference between the cumulative loss of the forecaster (that is, the sum of these negative logarithms) and that of the best expert in hindsight. The optimal strategy for sequentially assigning probability to outcomes is known to be normalized maximum likelihood (NML) [see, for e.g. [1], and [2], and see Definition 4 below]. NML suffers from two major drawbacks: the horizon  $n$  of the problem needs to be known in advance, and the strategy can be computationally expensive since it involves marginalizing over subsequences. In this paper, we investigate the optimality of two alternative strategies, namely the Bayesian strategy and the sequential normalized maximum likelihood strategy; see Definitions 5 and 6 below. Bayesian prediction under Jeffreys prior has been shown to be asymptotically optimal [see, for e.g. [2], chaps 7,8]. Moreover the regret of SNML is within a constant of the minimax optimal [3]. We show that for a very general class of parametric models (Definition 1), optimality of a Bayesian strategy means

that the strategy uses Jeffreys prior. Furthermore we show that optimality of the Bayesian strategy is equivalent to optimality of sequential normalized maximum likelihood. The major regularity condition for these parametric families is that the maximum likelihood estimate is asymptotically normal. This classical condition holds for a broad class of parametric models. The proofs and further details are in the full version of this paper [4].

## 2. DEFINITIONS AND NOTATION

We work in the same setup of [5] and use their definitions and notation. The goal is to predict a sequence of outcomes  $x_t \in \mathcal{X}$ , almost as well as a set of experts. We use  $x^t$  to denote  $(x_1, x_2, \dots, x_t)$ ,  $x^0$  to denote the empty sequence, and  $x_m^n$  to denote  $(x_m, x_{m+1}, \dots, x_n)$ . At round  $t$ , the forecaster's prediction is a conditional probability density  $q_t(\cdot | x^{t-1})$ , where the density is with respect to a fixed measure  $\lambda$  on  $\mathcal{X}$ . For example, if  $\mathcal{X}$  is discrete,  $\lambda$  could be the counting measure; for  $\mathcal{X} = \mathbb{R}^d$ ,  $\lambda$  could be Lebesgue measure. The loss that the forecaster suffers at that round is  $-\log q_t(x_t | x^{t-1})$ , where  $x_t$  is the outcome revealed after the forecaster's prediction. The difference between the cumulative loss of the prediction strategy and the best expert in a reference set is called the regret. The goal is to minimize the regret in the worst case over all possible data sequences. In this paper, we consider i.i.d. parametric constant experts parametrized by  $\theta \in \Theta$ .

**Definition 1 (Parametric Constant Model)** *A constant expert is an iid stochastic process, that is, a joint probability distribution  $p$  on sequences of elements of  $\mathcal{X}$  such that for all  $t > 0$  and for all  $x$  in  $\mathcal{X}$ ,  $p(x^t | x^{t-1}) = p(x_t)$ . A parametric constant model  $(\Theta, (\mathcal{X}, \Sigma), \lambda, p_\theta)$  is a parameter set  $\Theta$ , a measurable space  $(\mathcal{X}, \Sigma)$ , a measure  $\lambda$  on  $\mathcal{X}$ , and a parameterized function  $p_\theta : \mathcal{X} \rightarrow [0, \infty)$  for which, for all  $\theta \in \Theta$ ,  $p_\theta$  is a probability density on  $X$  with respect to  $\lambda$ . It defines a set of constant experts via  $p_\theta(x^t | x^{t-1}) = p_\theta(x_t)$ .*

For convenience, we will often refer to a parametric constant model as just  $p_\theta$ .

A strategy  $q$  is any sequential probability assignment  $q_t(\cdot | x^{t-1})$  that, given a history  $x^{t-1}$ , defines the condi-

tional density of  $x_t \in \mathcal{X}$  with respect to the measure  $\lambda$ . It defines a joint distribution  $q$  on sequences of elements of  $\mathcal{X}$  in the obvious way,

$$q(x^n) = \prod_{t=1}^n q(x_t | x^{t-1}). \quad (1)$$

In general, a strategy depends on the sequence length  $n$ . We denote such strategies by  $q^{(n)}$ .

**Definition 2 (Regret)** *The regret of a strategy  $q^{(n)}$  on sequences of length  $n$  with respect to a parametric constant model  $p_\theta$  is*

$$\begin{aligned} R(x^n, q^{(n)}) &= \sum_{t=1}^n -\log q_t^{(n)}(x_t | x^{t-1}) \\ &\quad - \inf_{\theta \in \Theta} \sum_{t=1}^n -\log p_\theta(x_t | x^{t-1}) \\ &= \sup_{\theta \in \Theta} \log \frac{p_\theta(x^n)}{q^{(n)}(x^n)} \end{aligned} \quad (2)$$

We consider a generalization of the regret of Definition 2. This is because some strategies are only defined conditioned on a fixed initial sequence of observations  $x^{m-1}$ . For such cases, we define the conditional regret of  $x^n$ , given a fixed initial sequence  $x^{m-1}$ , in the following way [see [2], chap. 11].

**Definition 3 (Conditional Regret)**

$$\begin{aligned} R^\Theta(x_m^n, q^{(n)} | x^{m-1}) &= \sum_{t=m}^n -\log q_t(x_t | x^{t-1}) \\ &\quad - \inf_{\theta \in \Theta} \sum_{t=m}^n -\log p_\theta(x_t | x^{t-1}) \\ &= \sup_{\theta \in \Theta} \log \frac{p_\theta(x^n)}{q^{(n)}(x_m^n | x^{m-1})} \end{aligned} \quad (3)$$

Notice that the strategy  $q^{(n)}$  defines only the conditional distribution  $q^{(n)}(x_m^n | x^{m-1})$ . We call such a strategy a conditional strategy. In what follows, where we consider a conditional strategy, we assume that  $x^{m-1}$  is such that these conditional distributions are always well defined.

**Definition 4 (NML)** *Given a fixed horizon  $n$ , the normalized maximum likelihood (NML) strategy is defined via the joint probability distribution*

$$p_{nml}^{(n)}(x^n) = \frac{\sup_{\theta \in \Theta} p_\theta(x^n)}{\int_{\mathcal{X}^n} \sup_{\theta \in \Theta} p_\theta(y^n) d\lambda^n(y^n)}, \quad (4)$$

provided that the integral in the denominator exists. For  $t \leq n$ , the conditional probability distribution is

$$p_{nml}^{(n)}(x_t | x^{t-1}) = \frac{p_{nml}^{(n)}(x^t)}{p_{nml}^{(n)}(x^{t-1})}, \quad (5)$$

where  $p_{nml}^{(n)}(x^t)$  and  $p_{nml}^{(n)}(x^{t-1})$  are marginalized joint probability distributions of  $p_{nml}^{(n)}(x^n)$ :

$$p_{nml}^{(n)}(x^t) = \int_{\mathcal{X}^{n-t}} p_{nml}^{(n)}(x^n) d\lambda^{n-t}(x_{t+1}^n). \quad (6)$$

The regret of the NML strategy achieves the minimax bound, that is,  $q^{(n)} = p_{nml}^{(n)}$  minimizes  $\max_{x^n} R(x^n, q^{(n)})$  [see, for e.g. [2] chap. 6]. Note that  $p_{nml}^{(n)}$  might not be defined if the normalization is infinite. In many cases, for a sequence  $x^{m-1}$  and for all  $n \geq m$ , we can define the conditional probabilities

$$p_{nml}^{(n)}(x_m^n | x^{m-1}) = \frac{\sup_{\theta \in \Theta} p_\theta(x^n)}{\int_{\mathcal{X}^{n-m+1}} \sup_{\theta \in \Theta} p_\theta(x^n) d\lambda^{n-m+1}(x_m^n)} \quad (7)$$

For these cases the conditional NML again attains the minimax bound, that is,  $q^{(n)} = p_{nml}^{(n)}$  minimizes  $\max_{x_m^n} R(x_m^n, q^{(n)} | x^{m-1})$  [see [2] chap. 11]. In both cases, the nml strategy is an equalizer, meaning that the regrets of all sequences of length  $n$  are equal.

**Definition 5 (SNML)** *The sequential normalized maximum likelihood (SNML) strategy has*

$$p_{snml}(x_t | x^{t-1}) = \frac{\sup_{\theta \in \Theta} p_\theta(x^t)}{\int_{\mathcal{X}} \sup_{\theta \in \Theta} p_\theta(x^t) d\lambda(x_t)}. \quad (8)$$

Notice that this update does not depend on the horizon. Under mild conditions, the regret of SNML is no more than a constant (independent of  $n$ ) larger than the minimax regret [3]. Once again,  $p_{snml}$  is not defined if the integral in the denominator is infinite. In many cases, for a sequence  $x^{m-1}$  and for all  $n \geq m$ , the appropriate conditional probabilities are properly defined. We restrict our attention to these cases.

**Definition 6 (Bayesian)** *For a prior distribution  $\pi$  on  $\Theta$ , the Bayesian strategy with  $\pi$  is defined as*

$$p_\pi(x^t) = \int_{\theta \in \Theta} p_\theta(x^t) d\pi(\theta). \quad (9)$$

The conditional probability distribution is defined in the obvious way,

$$p_\pi(x_t | x^{t-1}) = \frac{p_\pi(x^t)}{p_\pi(x^{t-1})}. \quad (10)$$

We denote the conditional Bayesian strategy for a fixed  $x^{m-1}$  as  $p_\pi(x_m^n | x^{m-1})$ .

Jeffreys prior [6] has the appealing property that it is invariant under reparameterization.

**Definition 7 (Jeffreys prior)** *For a parametric model  $p_\theta$ , Jeffreys prior is the distribution over the parameter space  $\Theta$  that is proportional to  $\sqrt{|I(\theta)|}$ , where  $I$  is the Fisher information at  $\theta$  (that is, the variance of the score,  $\partial/\partial\theta \ln p_\theta(X)$ , where  $X$  has density  $p_\theta$ ).*

Our main theorem uses the notion of exchangeability of stochastic processes.

**Definition 8 (Exchangeable)** *A stochastic process is called exchangeable if the joint probability does not depend on the order of observations, that is, for any  $n > 0$ , any  $x^n \in \mathcal{X}^n$ , and any permutation  $\sigma$  on  $\{1, \dots, n\}$ , the probability of  $x^n$  is the same as the probability of  $x^n$  permuted by  $\sigma$ .*



When we consider the conditional distribution  $p(x_m^n | x^{m-1})$  defined by a conditional strategy, we are interested in exchangeability of the conditional stochastic process, that is, invariance under any permutation that leaves  $x^{m-1}$  unchanged.

The asymptotic normality of the maximum likelihood estimator is the major regularity condition of the parametric models that is required for our main result to hold.

**Definition 9 (Asymptotic Normality of MLE)** Consider a parametric constant model  $p_\theta$ . We say that the parametric model has an asymptotically normal MLE if, for all  $\theta_0 \in \Theta$ ,

$$\sqrt{n} \left( \hat{\theta}_{(x^n)} - \theta_0 \right) \xrightarrow{d} N \left( 0, I^{-1}(\theta_0) \right), \quad (11)$$

where  $I(\theta)$  is the Fisher information at  $\theta$ ,  $x^n$  is a sample path of  $p_{\theta_0}$ , and  $\hat{\theta}_{(x^n)}$  is the maximum likelihood estimate of  $\theta$  given  $x^n$ , that is,  $\hat{\theta}_{(x^n)}$  maximizes  $p_\theta(x^n)$ .

Asymptotic normality holds for regular parametric models; for typical regularity conditions, see for example, Theorem 3.3 in [7].

For parametric models whose maximum likelihood estimates take values in a countable set, we need the notion of a lattice MLE.

**Definition 10 (Lattice MLE)** Consider a parametric model  $p_\theta$  with  $\theta \in \Theta \subseteq \mathbb{R}^d$ . The parametric model is said to have a lattice MLE with diminishing step-size  $h_n$ , if for any  $\theta$ , the possible maximum likelihood estimates of  $n$  i.i.d random variables generated by  $p_\theta$  are points in  $\Theta$  that are of the form  $(b + k_1 h_n, b + k_2 h_n, \dots, b + k_d h_n)$ , for some integers  $k_1, k_2, \dots, k_d$  and some real numbers  $b$  and  $h_n$ . Additionally  $h_n$  is positive and diminishes to zero as  $n$  goes to infinity.

We are now ready to state our main result.

### 3. MAIN RESULT

We show that in parametric models with an asymptotically normal MLE, the optimality of a Bayesian strategy implies that the strategy uses Jeffreys prior. Furthermore we show that the optimality of a Bayesian strategy is equivalent to the optimality of sequential normalized maximum likelihood. This extends the result for canonical minimal exponential family distributions from [5] to regular parametric models. Note that NML is the unique optimal strategy, so when we say that some other strategy is equivalent to NML, that is the same as saying that strategy predicts optimally.

**Theorem 3.1** Suppose we have a parametric model  $p_\theta$  with an asymptotically normal MLE. Assume that the MLE has a density with respect to Lebesgue measure or that the model has a lattice MLE with diminishing step-size  $h_n$ . Also assume that  $I(\theta)$ , the Fisher information at  $\theta$  is continuous in  $\theta$ , and that, for all  $x$ ,  $p_\theta(x)$  is continuous in  $\theta$ . Also fix  $m > 0$  and  $x^{m-1}$ , and assume that  $p_{nml}^{(n)}(x_m^n | x^{m-1})$  and  $p_\pi(x_m^n | x^{m-1})$  are well defined, where  $\pi$  is the Jeffreys prior. Then the following are equivalent.

(a) NML = Bayesian:

There is a prior  $\pi$  on  $\Theta$  such that for all  $n$  and all  $x_m^n$ ,

$$p_{nml}^{(n)}(x_m^n | x^{m-1}) = p_\pi(x_m^n | x^{m-1}) \quad (12)$$

(b) NML = SNML:

For all  $n$  and all  $x_m^n$ ,

$$p_{nml}^{(n)}(x_m^n | x^{m-1}) = p_{snml}(x_m^n | x^{m-1}) \quad (13)$$

(c) NML = Bayesian with Jeffreys prior:

If  $\pi$  denotes Jeffreys prior on  $\Theta$ , for all  $n$  and all  $x_m^n$ ,

$$p_{nml}^{(n)}(x_m^n | x^{m-1}) = p_\pi(x_m^n | x^{m-1}) \quad (14)$$

(d)  $p_{snml}(\cdot | x^{m-1})$  is exchangeable.

(e) SNML = Bayesian:

There is a prior  $\pi$  on  $\Theta$  such that for all  $n$  and all  $x_m^n$ ,

$$p_{snml}(x_m^n | x^{m-1}) = p_\pi(x_m^n | x^{m-1}) \quad (15)$$

(f) SNML = Bayesian with Jeffreys prior:

If  $\pi$  denotes Jeffreys prior on  $\Theta$ , for all  $n$  and all  $x_m^n$ ,

$$p_{snml}(x_m^n | x^{m-1}) = p_\pi(x_m^n | x^{m-1}) \quad (16)$$

## 4. OPEN PROBLEM

Our main result, i.e. Theorem 3.1 shows that the Bayesian strategy under Jeffreys prior, SNM and NML are all equivalent if and only if SNM is exchangeable. This equivalence holds for many exponential family distributions such as Normal, Levy, Rayleigh, Exponential. On the other hand it does not hold for some simple distributions such as Bernoulli. What properties should a distribution from an exponential family have that makes its sequential normalized maximum likelihood process exchangeable?

## 5. REFERENCES

- [1] Nicolo Cesa-Bianchi and Gabor Lugosi, *Prediction, Learning, and Games*, Cambridge University Press, New York, NY, USA, 2006.
- [2] Peter D Grunwald, *The Minimum Description Length Principle*, Cambridge, Mass. : MIT Press, 2007.
- [3] Wojciech Kotlowski and Peter Grünwald, “Maximum likelihood vs. sequential normalized maximum likelihood in on-line density estimation,” *Journal of Machine Learning Research - Proceedings Track*, vol. 19, pp. 457–476, 2011.
- [4] Fares Hedayati and Peter Bartlett, “The optimality of Jeffreys prior for online density estimation and the asymptotic normality of maximum likelihood estimators,” in *Proceedings of the Conference on Learning Theory (COLT2012)*, 2012, vol. 23, pp. 7.1–7.13.

- [5] Fares Hedayati and Peter Bartlett, “Exchangeability Characterizes Optimality of Sequential Normalized Maximum Likelihood and Bayesian Prediction with Jeffreys Prior,” *JMLR Workshop Conference Proceedings*, vol. 22: AISTATS 2012, pp. 504–510, 2012.
- [6] H. Jeffreys, “An invariant form for the prior probability in estimation problems,” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.
- [7] Whitney K. Newey and Daniel McFadden, “Chapter 35: Large sample estimation and hypothesis testing,” in *Handbook of Econometrics*, Robert Engle and Dan. McFadden, Eds., vol. 4, pp. 2111–2245. Elsevier Science, 1994.

# Author Index

- Adamskiy, Dimitri, [76](#)
- Bartlett, Peter L., [86](#)
- Bickel, David, [59](#)
- Buhman, Joachim M., [14](#)
- Busetto, Alberto Giovanni, [14](#)
- Dębowski, Łukasz, [59](#)
- E. García, Jesús E., [79](#)
- Eerola, Mervi, [10](#)
- Eggeling, Ralf, [32](#)
- García, Jesús E., [68](#)
- González-López, Verónica Andrea, [68](#), [79](#)
- Grosse, Ivo, [32](#)
- Haghir Chehreghani, Morteza, [14](#)
- Hedayati, Fares, [86](#)
- Helske, Satu, [10](#)
- Heusdens, Richard, [40](#)
- Hirai, So, [30](#)
- Ikeda, Shiro, [36](#)
- Kalnishkan, Yuri, [44](#)
- Kanazawa, Hiroki, [26](#)
- Knobbe, Arno, [64](#)
- Koolen, Wouter, [76](#)
- Liski, Antti, [72](#)
- Liski, Erkki P., [72](#)
- Mazumder, Anjali, [63](#)
- Myllymäki, Petri, [32](#)
- Nijssen, Siegfried, [64](#)
- Nouri, Javad, [52](#)
- Reshetnikov, Kirill, [52](#)
- Reznikova, Zhanna, [12](#)
- Roos, Teemu, [32](#)
- Ryabko, Boris, [6](#), [12](#)
- Ryabko, Daniil, [18](#)
- Sakurai, Ei-ichi, [26](#)
- Tatti, Nikolaj, [82](#)
- Topsøe, Flemming, [22](#)
- Van Ommen, Thijs, [4](#)
- Vanschoren, Joaquin, [64](#)
- Vespier, Ugo, [64](#)
- Viola, M. L. L., [68](#), [79](#)
- Vovk, Vladimir, [44](#), [48](#)
- Vreeken, Jilles, [82](#)
- Vyugin, Michael V., [44](#)
- Warmuth, Manfred K., [76](#)
- Watanabe, Kazuho, [36](#)
- Wettig, Hannes, [52](#)
- Yamanishi, Kenji, [26](#), [30](#)
- Yangarber, Roman, [52](#)
- Zhang, Guoqiang, [40](#)