

That Simple Device Already Used by Gauss

Peter Grünwald
CWI Amsterdam
the Netherlands
www.cwi.nl/~pdg

Abstract

From November 1998 until September 1999, Jorma Rissanen and I met on a regular basis. Here I recall some of our stimulating conversations and some of the work that we did together. This work, based almost exclusively on a single page of [12], was left unfinished and has never been published, but it has indirectly had a profound impact on my career.

1 Meet Jorma Rissanen

I first met Jorma in November 1998. I had just obtained my Ph.D. in Amsterdam and started a postdoc at Stanford University. These were exciting times: it was at the height of the dot-com boom, and Stanford was right in the middle of it. Since my thesis was all about the MDL Principle, I had suggested that Jorma and I could meet in person during my stay in California. Jorma replied that he would like to. I was delighted, honored but also a bit worried, since I had been forewarned that Jorma was not your “usual” kind of scientist...

San Francisco, Category Theory and Statistics I found that, while Jorma was not one for polite small talk, he did like having lots of beer with a small circle of academic friends (for most scientists it seems to be the other way around). He didn’t talk much, but what he said was invariably to the point, direct and frank. During our first meeting, when I told him that I lived in San Francisco since it seemed to me so much nicer than Palo Alto, his immediate reply was “I don’t like San Francisco”. Later that day, we talked about science in general, and I was quite surprised to learn that, in his first years at IBM, Jorma had been a serious student of category theory – he even published on it while he was professor in Sweden [15]. He went on to say that he found category theory to be *much* easier than statistics, the field to which he had made such major contributions. When I asked him why, he said “*because much of statistics is nonsense. It is exceedingly hard to teach yourself nonsense!*”. Vintage Jorma: blunt and sharp at the same time. Some fear him for this, others, like me, find Jorma’s conversation delightfully refreshing.

Jorma is, in fact, notorious for his strong opinions about sub-fields of mathematics – when a Ph.D. student once told him that he had spent a lot of time studying complex function theory, Jorma exclaimed (not entirely seriously, I believe) “*you’ve wasted your youth!*”

Jorma could be as harsh about his own work as he could be about others work – I vividly recall him saying “how could I have been so stupid” when I suggested that the central proof in [13] was much more difficult than was needed. When I told him that I did not quite understand another proof of his, in [11], he told me that he himself only understands it some of the time, “when I have one of my better days”. I am referring to the proof of what is perhaps his most well-known theorem: the central result of [11], which is an extension of the information inequality, but, from a statistical point of view, can also be interpreted as — as Jorma put it, now less modestly but entirely correctly — “a grand Cramér-Rao theorem.”

Soccer and Impounded Cars Our first meeting went quite well, and I was honored that Jorma asked me to visit him again. In the end, I visited him every 3-4 weeks during my year at Stanford - at the time, Jorma was still at IBM Almaden, in San Jose, just 40 minutes further down the highway. I usually arrived at 10 in the morning and stayed for the whole day. Between 12 and 2 however, Jorma would often leave to play soccer with a group of friends, something he did three times a week. At the time he was 67, but still very good at it: in his youth Jorma had seriously considered becoming a professional soccer player. In the end, he had decided to pursue his Ph.D. instead – as Jorma told me during one of my visits, he still doesn't know whether he has made the right choice. When Jorma went out for soccer, I used to have lunch at IBM's luxurious cafeteria, paid for by Jorma's card. On one occasion, my car had broken down, and Jorma told me that Nemo, one of his soccer friends, could probably arrange a good new car for me for as little as \$ 150: Nemo was a car dealer who took over impounded cars from the police if they hadn't been picked up for more than a year. I was fascinated: a brilliant scientist who always speaks his mind, played soccer at world-class level and counts impounded car dealers among his friends.

During my visits, Jorma and I also did some work together. Rather than actually working on a joint publication, we were both pursuing related but distinct ideas, always discussing our latest progress during our meetings. So what did we work on?

2 Prediction is Coding

I learned about the MDL Principle from the monograph “Stochastic Complexity in Statistical Inquiry” [12], the “little green book,” in which Jorma so eloquently puts forward the main ideas underlying the MDL Principle. In Chapter 2, Jorma notes that different research communities have a different understanding of the concept of a “model”. In statistics, a model is usually a family of probability distributions, for example, the Gaussian or normal family. In other fields such as pattern recognition and machine learning, a model is usually a family of deterministic hypotheses or *predictors*. For example, we may try to find a relationship between a variable X , taking values in some set \mathcal{X} , and a variable Y , taking values in \mathcal{Y} , by considering a “model” \mathcal{F} , which is really a family of deterministic functions $f : \mathcal{X} \rightarrow \mathcal{Y}$. f is then fit to the data $(x_1, y_1), \dots, (x_n, y_n)$, using, for example, the least squares criterion; concrete examples are given below. Jorma claims that, from an MDL perspective, there is no real distinction between the probabilistic and the deterministic type of model: they may both be viewed as defining *codes* or equivalently, description methods for the data. Statistical inference should then proceed by selecting the model (set of description methods) that leads to the shortest codelength of the data.

A Simple Device Already Used by Gauss How, then, should we associate models with codes? For probabilistic models this is obvious: the Kraft inequality tells us that for every probability distribution P , there exists a uniquely decodable code such that, for all outcomes x , the codelength of x is (essentially) equal to $-\log p(x)$, p being the mass function of P . The information inequality indicates that this particular code is the *only* reasonable code that one may want to associate with P [3, Chapter 3]. But what about deterministic predictors f ?

According to Jorma, we should map them to codes as follows. We first map each f to a conditional distribution p_f , defined in such a way that $-\log p_f(y | x)$ is an affine (linear with constant offset) function of the loss $L(y, f(x))$ that f makes when predicting y given x . We should then use the code with lengths $-\log p_f$. His remarks are worth quoting in full [[12], page 18; mathematical notation slightly adjusted, material between square brackets and emphasis added by myself]:

...The two views however can be reconciled by the *simple device used already by Gauss* for the distributions bearing his name. In fact, for any desired distance

measure $L(y_i, \hat{y}_i)$, [and any predictor f under consideration] define a density function

$$p_f(y_i | x_i) = K e^{-L(y_i, f(x_i))}, \quad (1)$$

where K is so chosen that p_f becomes a proper density function over the range of y_i . Taking the product of these, $p_f(y^n | x^n) = \prod_{i=1}^n p_f(y_i | x_i)$, over the observed data set, gives the desired conditional density function for sequences [...]

[For example] let the data consist of a binary sequence $y^n = y_1, \dots, y_n$. With some predictor \hat{y}_i as a function of the past observations, let $L(y_i, \hat{y}_i) = 0$ if the prediction is correct, i.e., if $y_i = \hat{y}_i$, else let $L(y_i, \hat{y}_i) = 1$. The desired criterion for the goodness of the predictors is the number of mispredictions in the observed sequence. Picking in (1) the base of the exponential as 2, we get $P(1 | \hat{y}_i) = K$ or $K/2$, depending on what the predicted symbol is. In either case, $P(0 | \hat{y}_i) = 1 - P(1 | \hat{y}_i)$, which makes $K = 2/3$. With this the number of mistaken predictions made differs only by a constant from the quantity $-\sum_i \log p(y_i | \hat{y}_i)$, which is seen to be an expression in terms of probabilistic model [and corresponds to the codelength of the data according to a particular uniquely decodable code] ...As another example, with $L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$, (1) defines a normal density function with mean \hat{y}_i and variance $1/2$. The induced normal density function for the data $p_f(y^n | x^n)$, as defined by its negative logarithm, is

$$-\log p_f(y^n | x^n) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \frac{n}{2} \ln \pi.$$

Again we see that the sum of the squared errors differs from the negative logarithm of a density function only by a constant.

This single page constitutes *all* Jorma writes about the unification of different types of models. When I first read it (in 1995) I was highly intrigued: one is promised here a new, fully general notion of “model”, encapsulating all previous ones, in which a model really becomes *a language that allows one to express particular properties of the data* [3]. This claim is bold, exciting, but only treated in the most sketchy of fashions. The examples that Jorma gave raise all kinds of questions. As will become clear, trying to answer these, and validating Jorma’s claim, has, to a large extent, shaped my own career.

An immediate question that comes to mind is: why use logarithm to the base 2 in the 0/1-loss example? And, relatedly, why use variance $1/2$ in the Gaussian (squared error) example? These seem to be arbitrary choices. They may be justifiable if we do have some additional *probabilistic* knowledge about the situation we are trying to model, e.g. that the errors are Gaussian with known variance $1/2$. But we often want to use predictors with squared error in cases where we hardly have any probabilistic knowledge; in particular, we usually do not even want to assume normality. Does the approach still work in such cases?

Optimality To phrase this question more precisely, for a fixed class of predictors \mathcal{F} , a fixed loss function of interest $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ and a fixed $\beta > 0$, for each $f \in \mathcal{F}$, define an associated conditional probability distribution $P_{f,\beta}$ identified by its mass function $p_{f,\beta}(Y|X)$, as follows:

$$p_{f,\beta}(y | x) \equiv \frac{1}{Z(\beta)} e^{-\beta L(y;f(x))} \quad (2)$$

where $Z(\beta) \equiv \sum_{y \in \mathcal{Y}} e^{-\beta L(y;f(x))}$ is a normalization factor. (2) is extended to several outcomes by taking product distributions. For the 0/1-loss, $Z(\beta) = 1 + e^{-\beta}$; for other loss functions $Z(\beta)$ may depend on f and x ; if \mathcal{Y} is not countable, $p_{f,\beta}$ becomes a density with respect to some fixed underlying measure and the summation in the definition of $Z(\beta)$ becomes integration. For example, for the squared loss, with the variable substitution $\sigma^2 = 1/(2\beta)$, (2) becomes a conditional normal density and $Z(\beta) = 1/\sqrt{2\pi\sigma^2} = \sqrt{\beta/2\pi}$. We see that

for every choice of $\beta > 0$, the codelength obtained by coding with p_f is an increasing affine function of the loss induced by f : for all $x^n \in \mathcal{X}^n$, all $y^n \in \mathcal{Y}^n$, we have

$$-\log p_{f,\beta}(y^n | x^n) = \beta \sum_{i=1}^n L(y_i, f(x_i)) + n \ln Z(\beta). \quad (3)$$

Thus, for each given sequence of data, and any two predictors f_1 and f_2 , f_1 better fits the data in terms of L if and only if $p_{f_1,\beta}$ better fits the data than $p_{f_2,\beta}$ in terms of likelihood. For fixed β , the likelihood will be maximized for $p_{\hat{f},\beta}$, where \hat{f} is the f that minimizes, among all $f \in \mathcal{F}$, the *empirical risk* $\sum_{i=1}^n L(y_i, f(x_i))$. In my thesis I called this the *optimality* property of the mapping (2): the optimal (best-fitting in terms of L) \hat{f} gets mapped to the optimal (best-fitting in terms of likelihood) \hat{p} . This is one of the main reasons why the property (3) is desirable for our mapping from deterministic f to probabilistic p .

Reliability The question that was asked above can now be rephrased as: is there a natural choice for β ? After two years, in 1997, I discovered that such a natural choice exists: we should simply *learn* β from the data, using some likelihood-based method such as maximum likelihood, MDL or Bayesian inference. The resulting β has a particularly useful property which may be called *reliability*. To explain this further, assume, for example, that we use two-part code MDL [3]. If one has only vague or no prior knowledge about what β should be, the straightforward thing is to first encode some $f \in \mathcal{F}$, then encode some $\beta \geq 0$, and then encode the data with the encoded $p_{f,\beta}$, using the combination (f, β) which minimizes the total two-part code-length. In this way, we can usually gain some additional compression of the data, which indicates that β captures some interesting property of the data. This is indeed the case, as is now shown. Note first that under any reasonable method for coding β , for large n , the encoded β will be close to the ML estimator $\hat{\beta}_f$ achieving $\max_{\beta} p_{f,\beta}(y^n | x^n)$, where f is the previously encoded predictor f . This ML estimator has a very special property, and this property is what makes learning β the natural thing to do. Namely, letting $E_{f,\beta}[\cdot]$ denote expectation under $p_{f,\beta}$, we find that, no matter what data (x^n, y^n) is observed, for any fixed f , we have

$$E_{f,\hat{\beta}_f}[L(Y, X)] = \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)). \quad (4)$$

To see this, simply differentiate the minus log-likelihood (3) with respect to β . The minimum is achieved if we set the derivative to 0, and this gives, for arbitrary y' ,

$$\sum_{i=1}^n L(y_i, f(x_i)) + n \frac{-\sum_y L(y, y') e^{-\beta L(y, y')}}{Z(\beta)} = 0, \quad (5)$$

from which (4) follows. Note that the $p_{f,\beta}$ are really conditional distributions for Y^n given X^n , so they only induce a conditional expectation of $L(Y, X)$, i.e. a function which maps each value for X to a corresponding expectation. However, this function is a constant: the expectation does not depend on X , and we may treat it like a single number, as in (4). In my thesis, I called (4) the *reliability* property of the mapping (2), since, for each $f \in \mathcal{F}$, the best-fitting $\hat{\beta}_f$ gives a “reliable” (unbiased in a very strong sense) indication of the performance of f on future data.

Entropification The reliability property indicates that it may be useful not only to learn f , but also to learn β from the data, and this suggests mapping \mathcal{F} to the set of distributions $\mathcal{P} = \{P_{f,\beta} | f \in \mathcal{F}, \beta \geq 0\}$, where $P_{f,\beta}$ is given by (2). In my thesis and in [6], I called this mapping the *entropification* of \mathcal{F} , a name which has hardly caught on (but see [7]). As long as we restrict β to be nonnegative, the likelihood will be jointly maximized for a distribution $p_{\hat{f},\hat{\beta}_f}$ such that \hat{f} is optimal in terms of empirical risk, and $\hat{\beta}_f$ is a reliable estimate of the

performance of \hat{f} . There is a slight problem in that for some loss functions, $\hat{\beta}$ may become negative, and then (3) indicates that the maximum likelihood will be achieved for some $p_{f,\beta}$ such that f has the *largest* rather than the smallest empirical risk within the set \mathcal{F} . The problem can be avoided by restricting the model \mathcal{P} to $\beta \geq 0$. In fact, it is not clear whether this is a “problem” at all, because negative $\hat{\beta}$ has a clear interpretation. In the classification case, if $\hat{\beta}_f < 0$, this means that one obtains smaller loss by predicting a 0 whenever f predicts a 1 and vice versa. Thus, f makes a mistaken prediction on more than half of the sample points, which means that it performs worse than random guessing.

One may doubt the practical relevance of the optimality and reliability properties (3) and (4), since, if \mathcal{F} is large, the ML estimator $p_{\hat{f},\hat{\beta}_f}$ may of course be prone to terrible overfitting and one might prefer other estimators instead. However, (3) and (4) immediately imply “expected” versions of the two properties, and these make the notions relevant for *any* likelihood-based inference method, including Bayes and MDL. Namely, fix any joint distribution P^* on $\mathcal{X} \times \mathcal{Y}$. Let \tilde{f} be the unique best predictor relative to P^* , i.e.

$$\tilde{f} := \arg \min_{f \in \mathcal{F}} E_{X,Y \sim P^*} [L(Y, f(X))],$$

and let \tilde{P} be the conditional distribution of the form (2) that minimizes conditional KL-divergence to P^* , i.e.

$$\tilde{P} := \arg \min_{P \in \mathcal{P}} D(P^* \| P),$$

where $D(P^* \| P) := E_{X \sim P^*} [D(P_{Y|X}^* \| P)]$ is the conditional KL divergence [5]. For simplicity we assume here that \tilde{f} and \tilde{P} exist and are unique; otherwise, we make no assumptions about P^* ; in particular, we do not assume that $P^* \in \mathcal{P}$. Then, as shown in my thesis, we have

$$\tilde{P} = P_{\tilde{f},\tilde{\beta}} \quad (\text{optimality}),$$

for some $\tilde{\beta} \geq 0$, and, if $\tilde{\beta} > 0$, then

$$E_{\tilde{P}} [L(Y, \tilde{f}(X))] = E_{P^*} [L(Y, \tilde{f}(X))] \quad (\text{reliability}) \quad (6)$$

For discussion about the case $\tilde{\beta} = 0$, see [6].

Now, as the sample size increases, *if a likelihood-based estimation method for \mathcal{P} converges at all*, it will converge to the \tilde{P} minimizing KL divergence to P^* [3]. Thus, the expectation-versions of the reliability and optimality-properties indicate that, if predictors are mapped to distributions using the “entropification” method (2), then, whenever estimation methods such as two-part or predictive MDL converge at all, they will (a) converge to the \tilde{P} that leads to the best predictions in terms of the user-supplied loss function of interest; and (b) give a consistent (asymptotically unbiased) estimate of how good the predictions of \tilde{P} really are. This seems exactly what is wanted from a mapping from deterministic predictors to description methods. I also found that earlier, [9] had explored a one-part code for the 0/1-loss based on essentially the same idea (averaging out β rather than encoding it). Finally, a seemingly disturbing aspect of Rissanen’s approach (fixed β), when applied to the 0/1-loss, was its apparent difference from an earlier approach by Quinlan and Rivest [10], who also tried to apply MDL in a deterministic classification context. I found that, if we allowed β to be determined by the data, then a simple application of Stirling’s approximation showed that the Quinlan and Rivest approach in fact became *equivalent* to Rissanen’s method after all. All this lead me to believe that “entropification” (i.e. extending the Gauss-Rissanen idea by allowing β to be learned from the data) was the right way to go.

Simple vs. Nonsimple Loss Functions Yet all was not well: as discussed in my Ph.D. thesis, entropification is surrounded by a myriad of slings and arrows. Here I concentrate on the most important one: the whole idea only works if the loss function is such that $Z(\beta)$ does not depend on f and x . In my thesis, I called such loss functions “simple”. Both

the 0/1- and the square loss functions are “simple”, but most other loss functions of practical interest are not “simple”. On our very first meeting, while praising the entropification idea in general, Jorma expressed doubts that it could be extended to such more general loss functions. For example, in a classification context, we may deal with an *asymmetric loss function*, which applies when predicting a 0 while the outcome should have been a 1 is much worse than vice versa. For example, $L_{\text{as}}(0, 0) = L_{\text{as}}(1, 1) = 0$, $L_{\text{as}}(0, 1) = 1$, $L_{\text{as}}(1, 0) = 10^6$. Loss functions such as this one are very important in practical applications such as deciding whether or not a certain drug should be administered to a patient. L_{as} is *not* simple, since $\sum_y e^{-\beta L(y, y')}$ depends on y' . In that case, the mapping (2) leads to values of $Z(\beta)$ depending on f and x , and the optimality property (3) is destroyed. We could try to save it by defining $Z'(\beta) = \sup_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} e^{-\beta L(y, y')}$, and using (2) with $Z(\beta)$ replaced by $Z'(\beta)$. The distributions in \mathcal{P} would then become defective (summing to less than one). Note that we will interpret our distributions as codes, and from such a coding perspective, there is nothing wrong with defective distributions in principle: via Kraft’s inequality, they still correspond to codes. However, the approach is still flawed, since when using defective distributions in this way, the optimality property is restored but the reliability property is lost! How can one generalize the idea so that both the optimality and the reliability property continue to hold? Jorma was doubtful that this could be done, and that his own bold claim “prediction is coding” could still be maintained for such nonsimple loss functions (interestingly, Phil Dawid, during my thesis defense two months earlier, had raised exactly the same doubts).

Feeling challenged by Rissanen and Dawid, I spent many an afternoon in the Stanford or San Francisco sun, trying to figure out a way to make “entropification” work for a larger class of loss functions. I felt that somehow it was possible. Each time I drove over to Jorma, I discussed my newest ideas on the topic, and gradually, I convinced him that my approach was feasible. Usually he would just listen, make some encouraging but general remarks, and then one day later send me an email, invariably starting with “Peter:”, followed either by a counterexample to my latest approach or some other profound issue.

The Importance of Being Brief Just as in conversation, Jorma’s emails are invariably brief and to the point. Here is an example dated February 1999: “Peter: The thing that’s missing in your lemma is only to show the convergence $\hat{\theta} \rightarrow \theta^*$, which permits you to replace the almost sure convergence of the sum simply by the entropy. Incidentally, if you have time next week we should meet. Jorma.”

My responses were always long. Similarly, Jorma’s MDL books are short, mine is very long. Jorma sticks to his own principle. I would love to do the same, but find that I lack the time: as Blaise Pascal has said “I have only made this letter longer because I have not had the time to make it shorter.”

In the end, with Jorma’s help, I found a partial and surprisingly simple solution to the nonsimple loss problem, which I’ll now describe. For sake of generality, we drop the restriction that the set of possible predictions coincides with the set of outcomes \mathcal{Y} . Thus, let \mathcal{A} be the set of possible predictions (\mathcal{A} stands for “acts” or “actions”, as decision theorists would prefer to call them), and let \mathcal{Y} be the set of possible outcomes to be predicted. Then a predictor f is a function $f : \mathcal{X} \rightarrow \mathcal{A}$, and a loss function is a function $L : \mathcal{Y} \times \mathcal{A} \rightarrow \mathbb{R}$. With each loss function we can associate its range

$$\mathcal{L} := \{l \in \mathbb{R} : l = L(y, a) \text{ for some } y \in \mathcal{Y} \text{ and } a \in \mathcal{A}\}.$$

For simplicity, we restrict ourselves to cases where \mathcal{L} is finite. For example, with $L = L_{\text{as}}$, the asymmetric loss defined before, we have $\mathcal{L} = \{0, 1, 10^6\}$.

Coding the Loss rather than the Data The central idea to make entropification work again is *to code the losses rather than the y-values*. Thus, we associate each f and β with a code for describing, for each i , the size of the *loss* $l_i := L(y_i, f(x_i))$, obtained when predicting y_i based on $f(x_i)$. Thus, rather than coding y_i given x_i using a code not

depending on f (as in the original entropification-approach), we now code l_i using a code which does depend on f .

To this end, for each $\beta \geq 0$, we define a mass function p_β on \mathcal{L} , simply by setting

$$p_\beta(l) := \frac{1}{Z(\beta)} e^{-\beta l} \quad (7)$$

where $Z(\beta)$ is given by

$$Z(\beta) \equiv \sum_{l \in \mathcal{L}} e^{-\beta l} \quad (8)$$

p_β is extended to sequences by independence, $p_\beta(l_1, \dots, l_n) = \prod_{i=1}^n p_\beta(l_i)$.

For given data x^n, y^n , given $f \in \mathcal{F}$ and $\beta \geq 0$, we now code the corresponding losses l_1, l_2, \dots, l_n using the code with lengths $-\log p_\beta$. Thus, for each x^n, y^n , we get codelength

$$-\log p_\beta(l^n) = \beta \sum_{i=1}^n L(y_i, f(x_i)) + n \ln Z(\beta). \quad (9)$$

(note again that p_β does not depend on f , but l^n does). (9) corresponds to (3) and shows that our mapping satisfies *optimality*. How about reliability? Let's fix some f and compute the derivative of (9) with respect to β , i.e. $(d/d\beta)(-\log p_\beta(l^n))$. A straightforward calculation analogous to (5) shows that *if* the derivative is 0 for some $\beta > 0$, then the likelihood achieves a maximum (the minus log likelihood is minimized) for this β , and for this β , we have

$$E_\beta[L] = \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)).$$

But this is just the reliability property (4) again. Further analysis reveals that an analogue of the expectation-versions of reliability and optimality also holds. Thus, by directly coding losses rather than outcomes we have recovered the essential properties of entropification for simple loss functions!

This approach does the trick, but it also raises a lot of questions: (a) what if the maximum likelihood is achieved at some $\beta < 0$? (b) can we still think of inference based on coding of the losses rather than the data as a form of the MDL principle? Philosophically, can there be some rationale for coding losses rather than data? Or should we somehow further adjust the approach such that, from what is encoded, the data y_1, \dots, y_n can always be recovered? More generally, the idea to code losses rather than data, which implies that the objects one actually wants to encode depend on the code one chooses to encode them, seems highly unusual! (c) How does the approach extend to continuous loss functions and loss functions with infinite domains? (d) does there exist a general formulation which subsumes both the "simple" approach (2) and the nonsimple approach described above? (e) How does the approach compare to the "aggregating algorithm" designed by Vovk and others [16, 8, 17, 1]? The aggregating algorithm can also be re-interpreted as mapping predictors/loss functions to probability distributions using (2), but rather than being learned from the data, β is chosen as a function of the loss function, and sometimes the sample size, in order to achieve good worst-case performance. Intriguingly, it turns out that Vovk's approach can only work for nonsimple loss functions – when Vovk deals with the squared error loss function, he restricts the range to $[-1, 1]$, and then $Z(\beta)$ becomes dependent on $f(x)$.

3 An Unfinished Tale

I found the solution sketched above in the final weeks of my stay in Stanford. A few months later I discovered how one can avoid the problems with $\beta < 0$, and I also discovered a more general approach subsuming the original entropification method, the method described above, and a third method which works for some continuous asymmetric loss functions. Still,

the important question (b) remained open, and I thought that I should only publish this work after having given more thought to it. Yet, until 2003, there were always more urgent things to work on, and I could not find the time for these thoughts. Then, in 2003, John Langford and I discovered that even with the simple 0/1-loss, applying MDL or Bayesian inference to an “entropified” model class \mathcal{P} can be inconsistent: there exist sets of 0/1-predictors \mathcal{F} such that two-part MDL or Bayesian inference based on the associated \mathcal{P} never converges [4, 5]. Note that the optimality and reliability properties indicate, as written below (6), that *if MDL estimation converges at all, it converges to the “right” \hat{P}* . The problem that Langford and I discovered is that in some cases, MDL estimation does not converge at all! Even though the best classifier $\tilde{f} \in \mathcal{F}$ has a small description length, as the sample size increases, MDL keeps selecting ever more complex classifiers, all of which are of much worse quality than the simple \tilde{f} . This seemed to be such a setback for the whole entropification idea, that I further postponed sorting out the details. Thus, my paper “Prediction *is* Coding, ” about entropification for general loss functions, has been left unfinished and has never been published. I do hope to finish it someday soon! (but I’ve been saying that for years). Of course, we could have made it into a joint project with Jorma, but his interests shifted as well – soon he was all into the Kolmogorov minimum statistic. He did publish, in 2003, a paper on normalized maximum likelihood for ‘simple’ nonlogarithmic loss functions, which was inspired by our many conversations at Almaden [14].

The Impact of Entropification It should be clear, that, although, in the end, we have not jointly published about it, Jorma’s thoughts about “the device already used by Gauss”, while fitting on a single page, have had a tremendous influence on my career. It was the basis for a large part of my Ph.D. thesis, of my first COLT paper [6], of the Bayes/MDL-inconsistency papers [4, 2, 5] – the latter having caused quite a stir among some Bayesian statisticians. In 2004, I was awarded a prestigious VIDI-grant by NWO, the Dutch science foundation. This award has made it possible to start what is rapidly becoming my own research group. The grant proposal was, in fact, all about entropification. I should really like to thank Jorma for that single page in his book!

References

- [1] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games*. Cambridge University Press, Cambridge, UK, 2006.
- [2] P. D. Grünwald. Bayesian inconsistency under misspecification, 2006. Presentation at the Valencia 8 ISBA Conference on Bayesian Statistics.
- [3] P. D. Grünwald. Prediction is coding, 2007. Manuscript in preparation.
- [4] P. D. Grünwald and J. Langford. Suboptimality of MDL and Bayes in classification under misspecification. In *Proceedings of the Seventeenth Conference on Learning Theory (COLT’ 04)*, New York, 2004. Springer-Verlag.
- [5] P. D. Grünwald and J. Langford. Suboptimal behavior of Bayes and MDL in classification under misspecification. *Machine Learning*, 2007. To appear.
- [6] P.D. Grünwald. Viewing all models as ‘probabilistic’. In *Proceedings of the Twelfth Annual Workshop on Computational Learning Theory (COLT ’99)*, 1999.
- [7] M.D. Lee. Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin and Review*, 15(1):1–15, 2008.
- [8] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- [9] R. Meir and N. Merhav. On the stochastic complexity of learning realizable and unrealizable rules. *Machine Learning*, 19:241–261, 1995.

- [10] J. Quinlan and R. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.
- [11] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14:1080–1100, 1986.
- [12] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, 1989.
- [13] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996.
- [14] J. Rissanen. Complexity of simple nonlogarithmic loss functions. *IEEE Transactions on Information Theory*, 49(2):476–484, 2003.
- [15] J. Rissanen and Bostwick F. Wyman. Duals of input/output maps. In E.G. Manes, editor, *Category Theory Applied to Computation and Control, Proceedings of the First International Symposium*, volume 25 of *Lecture Notes in Computer Science*, pages 204–208. Springer, 1975.
- [16] V.G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory (COLT' 90)*, pages 371–383, 1990.
- [17] K. Yamanishi. A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Transactions on Information Theory*, 44(4):1424–1439, 1998.