Using a sample-dependent coding scheme for two-part MDL

Jakob Verbeek

Using a Sample-dependent Coding Scheme for Two-Part MDL

Jakob J. Verbeek
email:jverbeek@wins.uva.nl
Universiteit van Amsterdam, Faculteit WINS

Abstract

In this paper we report on our investigation on how using sample-dependent coding schemes can lead to poor results in applying Rissanen's Minimum Description Length (MDL) principle [Ris89]. The MDL principle is one of the many known model selection methods in the field of 'machine learning', 'statistics' or 'inductive inference'. We analyze the experimental results presented in [KMNR97] and provide a method to avoid the overfitting. We do so by using a different coding scheme than in [KMNR97].

1 Introduction

Rissanen's Minimum Description Length (MDL) principle [Ris89] is one of the many known model selection methods in the field of 'machine learning', 'statistics' or 'inductive inference'. The general idea of MDL can be explained as follows: We want to model some regularities or dependencies in the data. These regularities in the data can be used to compress the data. Models describe regularities in the data. The more regularity of the data some model captures, the more it can compress the data. Therefore, the model that enables us to compress the data the most is the best model.

Two-Part MDL (TP-MDL) is one way to use this idea. What we do is describe the data in two parts, hence the name. First, we describe the model. Second, we describe the data *using* the model. The model describes the regularities in the data, the description of the data using the model codes the deviation of the data from these regularities.

In the following we will assume that models can be ordered by their complexity. For now think of model complexity as the number of parameters in a model.

Suppose we want to model some data generating system and we are given more and more examples of the behavior of the system. TP-MDL would typically start selecting simple models when the sample size is still small. As more and more examples become available TP-MDL usually selects more and more complex models, until it converges to some constant model complexity.[1] However, in the experiments described in [KMNR97] TP-MDL selected very complex models until some number m^* of examples had been reached. The model complexity of the selected models converged *very* rapidly if the number of examples was greater than m^*. The experimental results obtained for TP-MDL in [KMNR97] surprised us. We had two main questions: Would TP-MDL be applied correctly in [KMNR97]? Can we devise a coding scheme such that the performance of TP-MDL is improved?

2 Experiments

In this section we discuss the experiments we conducted to answer the second question. Furthermore, we give some arguments why we think one should not use sample-dependent coding schemes for TP-MDL.

2.1 Problem Domain

The interval function selection problem, as defined in [KMNR97], can be defined as follows. The data source is a function $f_d : [0,1] \to \{0,1\}$ that divides the interval $[0,1]$ into $d+1$ equal intervals, each labeled zero or one. We call such functions *interval functions*. The examples are drawn according to a uniform distribution on $X = [0,1]$. The data sequences consist of pairs $(x \in X, y \in Y = \{0,1\})$. The value $f_d(x)$ is zero in the first and all other odd intervals, it is one in all other intervals. The task of the inference algorithms is to output some interval function. Ideally the algorithm should output the model that models the data generating function the best. The training data is corrupted with some noise to make the setting more realistic.

2.2 Coding Schemes

Now, we give the coding scheme that was used in [KMNR97] to code the interval functions, we call this coding scheme $KMDL$. **KMDL** In [KMNR97] the set of possible parameter values is restricted to the x-data. This is motivated by the observation that the x-data is guaranteed to yield optimal parameter values. So in this coding scheme the data is not only used to *select* a hypothesis, the data is also used to *describe* a hypothesis. To code a model using this coding scheme, we first code the number of parameters t. Next, we code the index of the function in an enumeration of all interval functions with t parameters, such that the parameters are restricted to the x-data. That is: we code the index of the function in an enumeration of all subsets with t elements of the

[1] See [BRY97].

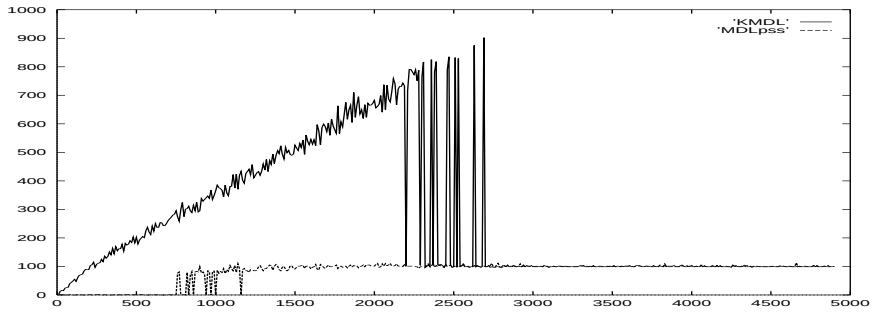

Figure 1: Plot of sample size, m, against hypothesis complexity, using a noise-rate of $\eta = 0.2$.

total set of all x-values in the training data.

Why should we use a sample-independent coding scheme to code the models? We give two arguments below, followed by the sample-independent coding scheme we used.

1. When we use a sample-dependent coding scheme to code the models, we do not describe a model that maps some arbitrary x-data to the Y-domain. What we obtain if we decode the codeword generated by $KMDL$ is a function that gives us a function $f : X \to Y$ if we provide a set of x-data points. Thus, $KMDL$ codes a function $f' : X^m \to (X \to Y)$, where X^m is the set of sets with m values from the X-domain. Usually, we use a function $f : X \to Y$, as these functions give us a dependency between X and Y, the dependency we want to model.
2. If we want to use the KMDL coding scheme to code a model (dependency between X and Y), we should provide the function f' and the x-data of the sample. So, KMDL provides us with an efficient way to code a model, *given* some *specific* x-data. However, KMDL is quite an inefficient way to code a model for some unspecified x-data set. So we can use the model to compress the observed data, but in general it will not enable us to compress some other data. We intuitively would like to have a model that allows us to compress any data (given that it is generated by the same data generating function).

Now, we give a sample-independent coding scheme to code models.

MDL$_{\mathbf{pss}}$ In this coding scheme we do not restrict parameter values to the x-data. First, we code the number of bits p, that is used to code one parameter value. Second, code the number of parameters t of the hypothesis. Third, code the index number of the function in the standard lexicographical enumeration of all functions with t parameters encoded using p bits precision.

To evaluate the performance of TP-MDL using these coding schemes we conducted several experiments.

3 Results and Conclusions

Using MDL_{pss} improves performance in two ways (see Figure 1): First, the overfitting that resulted from using $KMDL$ does not occur, instead underfitting occurs. Second, the number of examples needed to output the target *number* of parameters is considerably smaller when using MDL_{pss}.

However, using MDL_{pss}, the quality of *parameter estimation* decreases dramatically. With the quality of parameter estimation we mean how good the algorithm estimates the parameter values. Due to the minimization of code length the number of bits used by MDL_{pss} to code the parameters will be relatively small for small samples. The difference is even so dramatic that $KMDL$ outperforms MDL_{pss} in terms of generalization error. However, we can compensate for this by using MDL_{pss} only to select the *number* of parameters for the model and using Maximum Likelihood to select the actual parameter *values*. Further research should be carried out to find out whether this is in general a good method. A more elaborate report on this research is given in [Ver98].

References

[BRY97] A. Barron, J. Rissanen, and B. Yu. The Minimum Description Length Principle in Coding and Modeling. To be published in future, 1997.

[KMNR97] M. Kearns, Y. Mansour, A. Ng, and D. Ron. An Experimental and Theoretical Comparison of Model Selection Methods. *Machine Learning*, 27:7–50, 1997.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[Ver98] J. J. Verbeek. Overfitting using the MDL principle. Master's thesis, Universiteit van Amsterdam, Department of Mathematics, Computer Science, Physics, and Astronomy, 1998.