The Minimum Description Length Principle And Its Application to Online Learning of Handprinted Characters*

(Extended Abstract)

Qiong Gao and Ming Li

Department of Computer Science, York University
North York, Ont. M3J 1P3 Canada

ABSTRACT

Our objective is to introduce Rissanen's Minimum Description Length (MDL) Principle as a useful tool for character recognition and present a first application. Using MDL principle, a learning system has been implemented which, after simple training, is able to online recognize th trainer's handwriting in English and Chinese (including several thousand characters) with high success rate. The experimental results conform with the theoretical predictions. We will also try to give an elegant explanation of Rissanen's minimum description length principle (MDLP).

Areas: B2 -- Learning, knowledge acquisition; D1 -- Philosophical foundations.

1. Introduction

This research represents one of our efforts to apply the recently developed machine learning/inference theories [Valiant 1984, Rissanen 1978] to real world problems. In recent years, on one hand, new exciting learning theories have developed out from the computational complexity theory [V1], statistics and Kolmogorov complexity [R1]. These new theories received great attention in theoretical computer science and statistics. Plenty theoretical researches are done. See, for example, [V2, R2, R3, R4, R5, BEHW, KLPV]. One the other hand, these recent theoretical results are not yet applied to the real world learning system design with the exception of an elegant paper by Quinlan and Rivest [QR] (also independently by M. Wax). Our purpose is to try to bring theory and practice together and test some of the theories in real system designs. Specifically, we choose to apply the Rissanen's MDL Principle to design an on-line hand-written character learning system. Such a system has been implemented and its performance coincides with theoretical predictions.

One of the important aspects of AI research is the machine perception of natural human languages expressed in various ways. Enormous amount of effort has been made for recognition of handwritten characters or character strings [SU]]. Recognizing hand-written characters has applications, for example, in signature recognition and Chinese character input. In the latter case, there is a reasonable amount practical demands. This is because that there are several thousand independent Chinese characters; No current key-board input method is nearly natural enough for casual users; Some requires the user to memorize a code for each of seven thousand characters; Some requires the user to know *ping ying* and you still do not get what you type because there are too many homonyms; The sound recognition technique cannot help either since practically almost every commonly used Chinese character has more than one commonly used homonyms. For non-professional casual users, who do not want to spend time to actually *learn*, the hand-written input seems to be a quite reasonable choice.

A variety of approaches and algorithms have been used in order to achieve high recognition rate. The recognition process is usually divided into two steps: 1) feature extraction from the sample characters, and 2) classification of unknown characters. The latter often uses either deterministic or statistical inference based on the sample data and various different theories can be applied. However, for feature extraction, whose purpose is to capture the essence from the raw data, is largely in a state of art. Features such as center of gravity, moments, distribution of points, character loci, planar curve transformation, coordinates, slopes and curvatures at certain points along the character curve are the most commonly used ones. The obvious difficulty of recognition task is the variability involved in handwritten characters. Not only does the shape of the characters depend on the writing style which varies from person to person, even for the same person trying to write consistently, the difference of writing is noticeable from time to time. Therefore statistical and approximational approaches are usually used in order to deal with the variation. The elastic match is one of the technique successfully applied in this area (Kurtzburg, 1987 and Tapper, 1982). Briefly speaking, the elastic matching method takes the coordinates or slopes of certain points approximately equally spaced along the curve of the character drawing as feature to establish the character prototypes. To classify an unknown character drawing, the machine compares the drawing with all the prototypes in its knowledge base and the closest prototype is said to have the same character value as the unknown. When an unknown character is compared to a prototype, the comparison of the feature is not only made strictly between the corresponding

* This research was supported in part by ONR Grant N00014-85-K-0445, ARO Grant DAAL03-86-K-0171 at Harvard University, and by the NSERC operating grant OGP0036747 at York University.

point but also between the adjacent points in the prototype.

Presented with an example character, what features should we take? How many features should we take? Too few of them obviously cannot sufficiently describe the character; Too many of them would be make the algorithm too sensitive to noise and result worse recognition performance. For example, the above mentioned elastic matching uses certain points along the character curve as features. The interval used to extract these points along the curve is a parameter. How to determine this parameter? Practically speaking, we can set these interval to different values and experiment on the given sample data to see what value gives the best performance. However since the experiment is based on one particular set of data, we do not know if this interval value can give similar performance for all possible observations from the same data source. A theory is needed to guide the parameter selection in order to obtain the best description of data for future prediction.

Rissanen's MDL Principle serves this parameter selection purpose naturally. MDLP finds its root in the well known Bayesian inference and not so well-known Kolmogorov complexity. From the Bayes' rule, a specific hypothesis is inferred if the probability of the hypothesis takes maximum value for a given set of data and given prior probability distribution of a given set of hypotheses. When the Bayes' formula is expressed in the negative logarithmic form, the two terms, one is the conditional probability of the data for given hypothesis, and the other is the prior probability of the hypothesis, become the description length of the error given the hypothesis and the description length of the hypothesis respectively. Therefore, finding a maximum value of the conditional probability of a given set of hypotheses and data becomes minimizing the combined complexity or description length of the error and the hypothesis for a given set of candidate hypotheses.

From the viewpoint of data-compression used to encode a data set, the two description lengths are, in turn, expressed in terms of the coding lengths, i.e., the coding length of the hypothesis and the coding length of the error which is the part of data failed to be described by the hypothesis. They complement in the following way: if a hypothesis is too simple, it may fail to capture essence of the mechanism generating the data, resulting in bigger error coding lengths. On the other hand, if a hypothesis is too complicated and tends to include everything in the data, it may contain a lot of redundance from the data and become too sensitive to minor irregularities to give accurate predictions of the future data. The MDLP states that among the given set of hypothesis, the one with the minimum combined description lengths of both the hypothesis and the error for given set of data is the best approximation of the mechanism behind data and can be used to predict the future data with best accuracy.

The objective of this work is to implement a system of handprinted (English and Chinese) character recognition based Rissanen's MDLP. Specifically, MDLP is used in the selection of the interval of feature extraction. The result is tested experimentally to validate the application of the theory. The next section should serve as an elementary and practical introduction of the Rissanen's MDLP. Then in the following sections we apply this principle to our learning system.

2. The Rissanen's MDL Principle and Related Theories

Scientists formulate their theories in two steps: first a scientist must, based on scientific observations or given data, formulate alternative hypotheses (generally there are an infinity of alternatives), and second he selects one definite hypothesis. Historically this was done by many different principles, among the most dominant in statistics, the Fisher's Maximum Likelihood principle, various ways of using Bayesian formula (with different prior distributions). Among the most dominant in "common sense", the so-called Occam's razor principle of choosing the simplest consistent theory. However, no single principle is both theoretically sound and practically satisfiable in all situations. Fisher's principle ignores the prior probability distribution (of hypotheses). The application of Bayes' rule is hard usually due to unknown prior probability distribution. In order to resolve the problem of prior distributions, Solomonoff [S] and then Rissanen [R] proposed the following approach, which resulted the MDLP principle. By Bayesian rule we have:

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}, \tag{1}$$

where $P(H \mid D)$ is called the *final*, or *a posteriori*, probability, $P(H)$ is called the *initial*, or *a priori*, probability, and $P(D \mid H)$ is the conditional probability of seeing D when H is true. The posterior probability $P(H \mid D)$ is obtained by modifying the prior probability $P(H)$ according to the Bayes' rule (1). The Bayesian approach tells us to use the hypothesis H such that $P(H \mid D)$ is maximized. Since $P(D)$ can be considered as a normalizing factor, we ignore it in the following discussion. Now take the negative logarithm on both sides of the Bayesian formula (1), we get

$$-\log P(H \mid D) = -\log P(D \mid H) - \log P(H) + \log P(D) \tag{2}$$

Since we are only concerned with maximizing the term $P(H \mid D)$ or, equivalently, *minimizing* the term $-\log P(H \mid D)$, this is equivalent to minimizing

$$(-\log P(D \mid H)) + (-\log P(H)) \tag{3}$$

Now to get the minimum description length principle, we only need to explain the two terms in (3) correctly. First notice that both $P(D \mid H)$ and $P(H)$ are probabilities, so each is less than or equal to 1. Hence, $-\log P(D \mid H)$ and $-\log P(H)$ are positive numbers.

We explain $-\log P(H)$ first. $P(H)$ is so-called prior probability for hypothesis H to be true. Usually this is unknown, we can only have an initial estimate. The major issue here is to reasonably approximate it. In the original Solomonoff approach [S], H in general denotes a Turing machine. In practice we must avoid such too general approach in order to keep things computable. In different applications, the hypothesis H can mean many different things. For example, if we infer decision trees, H is a decision tree [QR]; In case of learning finite automata, H can be a finite automaton; In case we are interested in learning Boolean formulae, then H may be a Boolean formula; If we are fitting a polynomial curve to a set of data, then H may be a polynomial of some degree; In our case, H will be the model for a particular character. Each such H can be encoded by a binary string from a prefix-free set, where a set of codes is *prefix-free* if no code in the set is a prefix of another. Solomonoff suggested that we assign H the prior probability $2^{-K(H)}$ where $K(H)$ is informally the length of shortest prefix-free description of H, then $-\log P(H)$ is precisely the *length* of a minimum *prefix code* of the hypothesis H. More precisely $K(H)$ is the so-called self-delimiting Kolmogorov complexity of H, we will not get into this subject since it does not affect the reader's ability to understand the main issue. Interested readers are referred to [LV] for an introduction and more details and references of Kolmogorov complexity. In this case, by Kraft's inequality, $\sum_{H_i} 2^{-K(H_i)} \le 1$,

hence such probability assignment is reasonable. The optimality of this probability assignment is also proved in the sense that this probability assignment approximates any other *computable* probability distribution. However it is known that $K(H)$ is not computable. Such an approach can hardly be practical. Rissanen suggested the following approach to approximate the Solomonoff idea. First convert (or encode) H to an integer belonging to $Z=\{0,1,2,\cdots\}$. Then we try to assign prior distribution to each integer in Z. Jeffreys [J] suggested to give $\frac{1}{n}$ to integer n; but this results an improper distribution since $\sum_{n=1}^{\infty}\frac{1}{n}$ diverges. Rissanen suggested to assign number n with prior probability $2^{-L(n)}$ for

$$L(n)=log^*(n)+\log(c)$$

where $\log^*(n)=\log n+\log\log n+\log\log\log n+\cdots$, the sum including all the positive iterates, and $c=2.865064\cdots$. We get following desired properties: (1) $\sum_{n=1}^{\infty}2^{-L(n)}=1$ and (2) integer n is coded by a prefix code. Hence, descriptions of two integers, n_1 and n_2, can be just concatenated to produce the code for both n_1,n_2.

The first term $-\log P(D\mid H)$, also known as the *self-information* in information theory and the negative log likelihood in statistics, can be regarded as the number of bits it takes to redescribe or encode D with an ideal code relative to H. In fact one can show the following fact by Gibb's theorem from information theory:

Fact. Only a vanishing fraction of long strings can be encoded with appreciably fewer bits than $-\log P(D\mid H)$. And this ideal is reachable within a small number of excess bits by some suitable coding.

To summarize, we obtain the Rissanen's **Minimum Description Length Principle** which is intuitively stated as follows:

The best theory to explain a set of data is the one which minimizes the sum of

(1) the length (encoded in binary bits) of the description of the theory, *i.e.* the term $-\log_2 P(H)$;

(2) the length (in binary bits) of encoding of data with the help of the theory, *i.e.* the term $-\log_2 P(D\mid H)$, the error coding length.

See [LV] for more related discussions.

3. Learning System Development

3.1. Basic Assumptions

When a character is drawn on a planar surface, it can be viewed as a composite planar curve, the shape of the curve is completely determined by the coordinates of the sequence of points along the curve. The order of the sequence is determined from the time of writing the character. Obviously, the shape tends to vary from person to person and from time to time, so do the coordinates of the point sequence. A key assumption here is that for a particular person writing consistently, the shape of the curve tends to converge to a mean shape, the mean of coordinates of the point sequence converge respectively to some set of values. The probability distribution for the coordinates is left unknown, but is assumed to be symmetric about the mean values, and the variance is assumed to be the same for all the character drawings. Theoretically, we only assume that a *fixed* probability distribution of one person's hand-writing does not change. See [V].

3.2. Feature Space, Feature Extraction and Prototypes

A Kurta ISONE digitizer tablet with 200/inch resolution in both horizontal and vertical directions is used as the transducer to send the coordinates of the sequential points of the character curve on the tablet to the microprocessor of a IBM PS/2 model 30 computer. The system is implemented using programming language C. The coordinates are then standardized against 30 in both horizontal and vertical directions. The sequence of the coordinates becomes a linked list, which goes through preprocess in order to remove the repeated points due to hesitation at the time of writing, and to fill in the gaps between sampled points resulted from the sampling rate limit of the tablet. The latter needs some explanation: the digitizer has a maximum sampling rate of 100 points/second. If a person writes a character in 0.2 second, only 20 points on the character curve will be sampled, leaving gaps between those points. The preprocess procedure is to ensure that any pair of consecutive points on the curve after preprocessing have at least one component of coordinates, stored as integers, differing by 1 and no coordinate component differing greater than 1. The preprocessed curve coordinate list is then sent to feature extraction. So far the coordinates are still integers in the range of 0 to 30.

The coordinates of certain points along the character curves are taken as features. Feature extraction is done as follows: a character may consist of more than one stroke (a stroke is the trace from a pen drop-down to pen lift-up), the starting and ending points of every stroke are mandatorily taken as features. In between, feature points are taken at a fixed interval, say, one point for every n points along the preprocessed curve, where n is called **feature extraction interval**. This is to ensure that the feature points are roughly equally spaced. Actually the length between any two points, excluding the last point of a stroke, varies from n to $\sqrt{2}n$ ($\sqrt{2}n$ for the diagonal). All the feature point coordinates of a given character drawing constitute a feature vector whose dimension is $2n$ since every point has two coordinate components. Obviously the dimension of the feature vector is also a random variable since the shape and the total number of points on the character curve varies from time to time. The dimension of the feature vector is largely determined by the feature extraction interval.

Note, for Chinese characters, other features are also extracted and a decision tree is formed too speedup the process. These will be ignored at present abstract.

The extracted feature vector of a character can also be viewed as a prototype of the character and saved as the knowledge of the system.

3.3. Comparison between Feature Vectors

Before the system is employed to recognize characters, it has to be trained with the character drawings from the same data source which it is supposed to work with. Here the "same data source" means the same person writing consistently. The basic technique used in both training and recognition is the comparison or matching between prototypes or feature vectors. To compare any two prototypes or feature vectors, we simply take the absolute distance between the two vectors. Mathematically this means subtracting each component of one vector from its corresponding component in the other feature vector, summing up the square of the differences and taking the square root of the sum. *I.e.*

$$\sqrt{\sum_{i=1}^{n}(x_i-x'_i)^2+\sum_{i=1}^{n}(y_i-y'_i)^2}.$$

The comparison technique used here follows the spirit of this mathematical definition but is **elastic**. The elasticity is reflected in two aspects: First, the comparison is allowed for feature vector with different dimensions in the range of T_d, the **dimension tolerance**. The dimension tolerance is defined such that if the one character' s feature vector has n feature points, it will be compared to all the prototypes with feature vectors with number of feature points in the range of $[n-T_d, n+T_d]$. Second, the **local extensibility** N_e is allowed so that the i th feature point of one character feature vector is compared with the feature points of the other vector with index ranging from $i-N_e$ to $i+N_e$, the smallest difference is considered to be the "true" difference between the two vectors at *ith* feature point. The sum of the square of these "true" difference is considered to be the absolute difference between the two feature vector. This comparison technique is often referred as elastic matching. For our particular problem, both T_d and N_e are set to 1 based on the experience.

3.4. Knowledge Base and Learning

The knowledge base of the system is a collection of prototypes saved in the form of a linked list during the learning process. The establishment of the knowledge base follows the following rules:

1) When the system is called to accept a new character drawing with known character value, the system checks in its existing knowledge base to see if it has any prototypes existing for that character. If there is no existing prototype for a specific character, the newly arrived feature vector for that character will be saved as the first prototype for that character.

2) If there is at least one prototype for the character existing in the knowledge base, the newly arrived character feature vector will be compared selectively with the prototypes in the knowledge base whose number of feature point differs at most by T_d from that of the new character feature. The minimum absolute distance may or may not be one of the prototypes with the same character value. The new character prototype is then handled as the following:

i) If the minimum distance, defined as $D_{\min}$, is between the new character and one of the prototypes in the knowledge base with the same character value, the new prototype will be combined with the existing prototype by taking the weighted average of every the coordinate components to produce a modified prototype for that character.

ii) If the minimum distance is between the new character and one of the prototype in the knowledge base with different character value, but the distance between the new character and its closest prototype of the same character value in the knowledge base, defined as D_{minc}, is not greater than $D_{\min}(m+1)/m$, where m is number of character drawings combined in the way described above to produce the closest prototype of the same character value in the knowledge base, the new character prototype will still be combined with the closest prototype of the same character value to provide a modified prototype. It is expected that the modified prototype will be able to assume $D_{\min}$ next time when a similar drawing of the same character value is arrived.

iii) Otherwise the new prototype will be saved in the knowledge base as a new prototype for the character. Therefore more than one prototype for a single character may exist in the knowledge base.

3.5. Recognition of an Unknown Character Drawing

When an unknown character arrives at the system, the system will compare it to all the prototypes in the knowledge base with dimension variation within the range specified by the dimension tolerance. The character of the prototype which has minimum absolute distance from the unknown is considered to be the character value of the unknown, since the prototypes are the "mean" values of the feature vectors of the characters and the variances of the distribution are assumed to be the same.

This concludes the main procedure of training and classification. A few remarks should be made here:

A. This process differs from the original elastic matching method in the the way of prototype construction. More than one prototypes are allowed for a single character and a prototype is the statistical mean of a number of positive examples of the character.

B. Every prototype is a feature vector which in turn is a point in the feature space of its dimension. Since the classification is based on statistical inference, the rate of correct classification depends not only on how well the prototypes in the knowledge base are constructed, but also the variability of the handwriting of a person. Even though more than one prototypes are allowed for any character in the knowledge base, too many prototypes may result in over-densified feature space and when the absolute distance between two points (two prototypes in the knowledge base) in the feature space is comparable to the variability of writing, the rate of correct classification may be considerably decreased.

C. The prototypes in the knowledge base constitute the hypothesis for the system. How well the prototypes are constructed will essentially determine the rate of correct classification, therefore the performance of the hypothesis. For the scheme described above, the prototypes is constructed by extracting points at a constant interval. Generally speaking, the more points in the prototypes gives more detailed image about the character drawing but may also include some random "noise" component in the hypothesis. Application of Rissanen's MDLP to guide the selection of "best" feature extraction interval is what is mainly aimed at by this work.

4. Description Lengths and Minimization: Experimental Results

The expression in Rissanen's MDLP consists of two terms: the hypothesis and error coding lengths. The coding efficiency for both of this two terms must be comparable, otherwise minimizing the resulted expression of total description length will give either too complicated or too simple hypotheses.

Although the whole system is implemented and it is able to recognize several thousands of characters now, in order to make our point clear, we will describe our experiments over 62 alphanumerals: 0,...,9,A,...,Z,a,...,z.

For this particular problem, the coding lengths are from the practical programming consideration. A set of 186 character drawings, exactly 3 for each of the 62 alphanumeral characters, were recorded in a raw database. The character drawings were stored in standardized integer coordinate system ranged from 0 to 30 in both x and y direction. These character drawings were then input to the system to establish a knowledge base, which was the collection of prototypes with normalized real coordinates, based on a selected feature extraction interval. After the construction of knowledge base was finished, the system was tested by having it classify the same set of character drawings. The error coding length is the

sum of the total number of points for all the incorrectly classified character drawings and the hypothesis coding length is the total number of points in all the prototypes in the machine's knowledge base multiplied by 2. The factor of 2 is from the fact that the prototype coordinates are stored as real numbers which takes twice as much memory (in C) as the character drawing coordinates which is in integer form. One might wonder why the prototype coordinates are real instead of integer numbers. The reason is to facilitate the elastic matching to give small resolution for comparisons of classification.

Thus both the hypothesis and error coding lengths are directly related to the feature extraction interval. The smaller this interval, the more complex the hypothesis, but the smaller the error coding length. The effect is reversed if the feature extraction interval goes toward larger values. Since the total coding length is the sum of the two coding lengths, there should be a value of feature extraction interval which rends the total coding length a minimum. This feature extraction interval is considered to be the "best" one in the spirit of MDLP and the corresponding model, the knowledge base, is considered to be optimal in the sense that it contains enough essence from the raw data but eliminates most redundance of noise component from the raw data. This optimal feature extraction interval can be found by carrying out the above described build-and-test (buiding the knowledge base then test it based on the same set of characters on which it was built.) for a number of different extraction interval.

The actual optimization process is implemented on the system and is available whenever the user wants to call. For our particular set of characters, the results of optimization is given in Figure 1, which depicts three quantities: the hypothesis, the error and the total coding lengths versus feature extraction interval (SAMPLING INTERVAL int the Figure). For larger feature extraction interval, the hypothesis complexity is small but most of the character drawings are misclassified, giving the very large total coding length. On the other hand, when the feature extraction interval is at its small extremity, all the training characters gets correctly classified, thus the error coding length is zero. However the hypothesis complexity reaches its largest value, resulting in a larger total coding length also. The minimum coding length occurred at extraction interval of 8, which gives 98.2 percent of correct classification. Figure 2 illustrates the fraction of correctly classified character drawings for the training data.

5. Validation of the Hypothesis

Whether the resulted "optimal" hypothesis really performs better than the hypotheses in the same class, the knowledge bases established using different feature extraction intervals, is subject to test by new data of character drawings. For this testing purpose, three sets of 62 characters were drawn by the same person who provided the raw data base to build the knowledge base. Thus the new data is considered to be from the same source as the previous data set. This new data set is classified by the system using the knowledge bases built from the former data set of 186 character drawings, based on different feature extraction intervals. The testing result is plotted in Figure 3 in terms of the fraction of correct classification (CORRECT RATIO) versus feature extraction interval. It is interesting to see that 100% correct classification occurred at feature extraction intervals 5, 6 and 7. These values of feature extraction intervals are close to the optimized value 7-8. Furthermore, at the lower end of feature extraction interval, the correct classification drops down, indicating the disturbance of too much redundance in the hypothesis. The recommended working feature extraction interval is thus 7-8 for this particular type of character drawings.

6. Summary, related results, and future research

Rissanen's Minimum Description Length Principle is applied to handprinted character recognition using elastic matching and statistical technique. The hypothesis is a collection of prototypes built from raw character drawings by taking points on the curves of character drawing at a constant feature extraction interval and by combining closely related character drawings. The hypothesis is optimized in the spirit of MDL principle by minimizing the total coding length which is the sum of the hypothesis and error coding lengths against feature extraction interval. The resulted hypothesis is tested using a different set of character drawing from the same source. The result of test indicates that MDLP is a good tool in the area of handprinted character recognition.

The following related work were brought to the authors' attention: Stanfill (AAAI-87), Bradshaw (MLW-87), Aha and Kibler (MLW-87), and Gennari, Langley and Fisher [GLF]. We plan to discuss these related work in the final version. Currently our experiment is obvious very preliminary. We plan to perform more experiments with more data and with different methods, for exmaple with different interval lengths as suggested by Pat Langley.

ACKNOWLEDGEMENT

We are grateful to Les Valiant for many discussions on machine learning and the suggestion of this research. We are also grateful to the vey helpful suggestions of Pat Langley and several referees whose suggestions will be implemented in the final version of this paper.) We also thank Paul Vitanyi and Mati Wax for useful discussions on MDL principle.

REFERENCES

[BEHW] Blumer, A, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Classifying Learnable Geometric Concepts With the Vapnik-Chervonenkis Dimension." ACM STOC, pp. 273-282. 1986.

[GLF] J. Gennari, P. Langley, D. Fisher, Models of incremental concept formation, UC. Irvine, TR-88-16.

[H] Jeffreys, H., "Theory of Probability", 3rd Edition. Clarendon Press, Oxford, 1961.

[KLPV] Kearns, M., M. Li, L. Pitt, and L.G. Valiant, "On the Learnability of Boolean Formulae", *19th ACM Symposium on Theory of Computing*, pp. 285-295, 1987.

[K] Kurtzberg, J., "Feature Analysis for Symbol Recognition by Elastic Matching" IBM J. Res. Develop., 1987, 31:1, (91 - 95).

[LV] Li, M, and P. Vitanyi, "Two decades of Applied Kolmogorov Complexity", *Proc. 3rd Structure in Complexity Theory*, pp. 80-101. Also to appear in *Handbook of Theoretical Computer Science*, (J. van Leeuwen, Managing Editor).

[QR] Quinlan J.R. and R. Rivest, "Inferring Decision Trees Using the Minimum Description Length Principle", Preprint, LCS, MIT, 1987.

[R1] Rissanen, J., "Modeling by Shortest Data Description", *Automatica*, 14, pp. 465-471, 1978

[R2] Rissanen, J., "Stochastic Complexity and Modeling", The Annual Statistics, 1986, 14:3 (1080 - 1100).

[R3] Rissanen, J., "Universal Coding, Information, Prediction, and Estimation" IEEE Transactions on Information Theory, 1984, 30:4, (629 - 636).

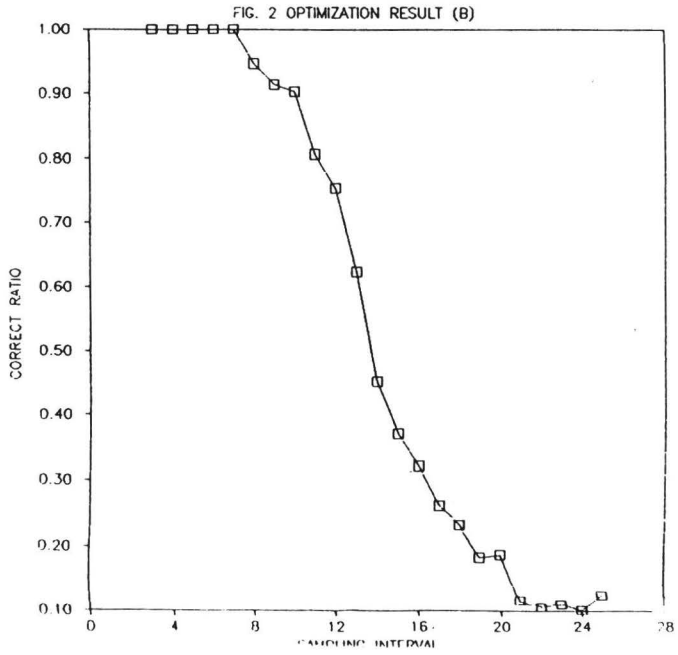
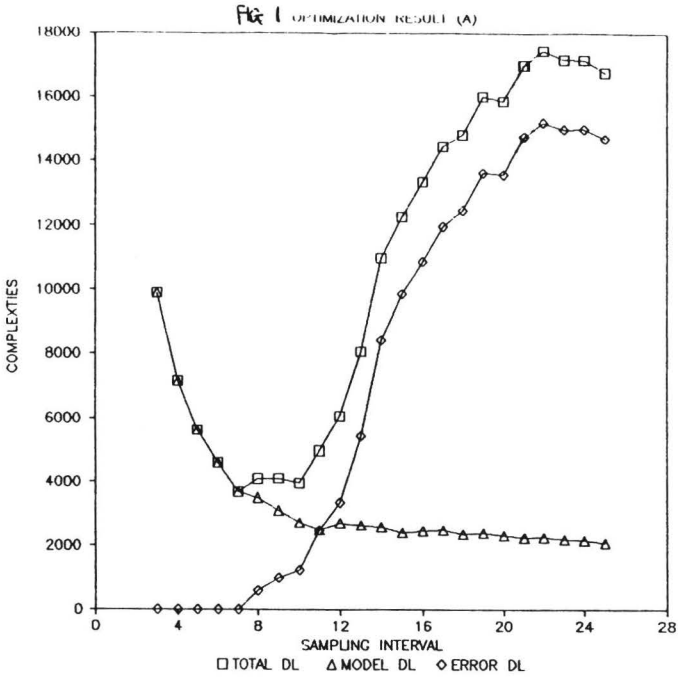
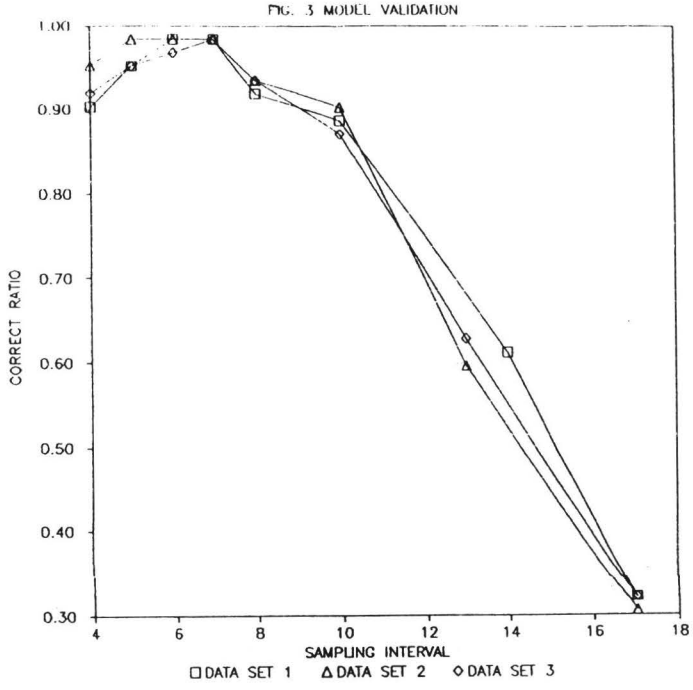
[R4] Rissanen, J., "Stochastic Complexity", *J.R. Statist. Soc.*, 49,3, pp 223-239 and 252-265, 1987

[R5] Rissanen, J., "Minimum Description Length Principle", *Encyclopedia of Statistical Sciences*, 5 (S. Kotz and N.L. Johnson, eds.), pp. 523-527, New York, Wiley, 1985.

[S] Solomonoff, R., "A Formal Theory of Inductive Inference. Part 1, *Information and Control*, 7, pp. 1-22, Part 2, 7, pp. 224-254. 1964

[SU] Suen, C. Y., "Automatic Recognition of Handprinted Characters -- The State of Art" Proceedings of the IEEE, 1980, 68:4, (469 - 487).

[T] Tapper, C. C., "Cursive Script Recognition by Elastic Matching" IBM J. Res. Develop., 1982, 26:6 (765 - 771).

[V1] Valiant, L.G. "A Theory of the Learnable", *Comm. ACM*, 27, 11, pp. 1134-1142, 1984.

[V2] Valiant, L.G. "Deductive Learning", *Phil. Trans. R. Soc. Lond.* A 312, 441-446, 1984.