Inductive Learning in Probabilistic Domain

Yoichiro Nakakuki Yoshiyuki Koseki Midori Tanaka
C&C Systems Research Laboratories, NEC Corp.
4-1-1 Miyazaki, Miyamae-ku, Kawasaki 213
nakakuki%asl.cl.nec.co.jp@uunet.UU.NET
JAPAN

Abstract

This paper describes an inductive learning method in probabilistic domain. It acquires an appropriate *probabilistic model* from a small amount of observation data. In order to derive an appropriate probabilistic model, a *presumption tree* with least description length is constructed. Description length of a presumption tree is defined as the sum of its code length and log-likelihood. Using a constructed presumption tree, the probabilistic distribution of future events can be presumed appropriately from observations of occurrences in the past. This capability enables the efficiency of certain kinds of performance systems, such as diagnostic system, that deal with probabilistic problems. The experimental results show that a model-based diagnostic system performs efficiently by making good use of the learning mechanism. In comparison with a simple probability estimation method, it is shown that the proposed approach requires fewer observations, to acquire an appropriate probabilistic model.

1 Introduction

In recent years, there has been a growing amount of research on inductive learning. Most of works has focused on deterministic domains, rather than uncertain ones. In a deterministic domain, we can induce deterministic classification rules, such as a decision tree, from given examples. On the other hand, in a probabilistic domain, we can only presume a probabilistic distribution(a probabilistic model) based on the observed occurrence of events. Although several approaches[Mingers, 1989; Quinlan, 1986; Yamanishi, 1989] deal with uncertainty, they are concerned with predicting a class, not the probabilistic distribution. The objective of this paper is to develop a mechanism to induce a probabilistic distribution in such a domain.

In general, a performance system which deals with probabilistic problems, such as malfunctions of a device, must incorporate experiential knowledge about the probabilistic distribution. For example, a model-based diagnostic system without any heuristics requires many tests to pinpoint a failing component. Therefore, some heuristics on the probability, such as 'most-probable-first heuristics' are indispensable[de Kleer and Williams, 1987; de Kleer and Williams, 1989; Koseki, 1989].

However, it is not easy to induce an appropriate probabilistic model from observed data. Especially, if the number of observed data is small, quite different models may become candidates for the selection. Therefore, an appropriate criterion for the model selection is indispensable.

In this paper we introduce a *presumption tree* to describe a probabilistic model. Using a presumption tree, we can presume the probability of each event. To obtain the most appropriate presumption tree for given observation data, the minimum description length(MDL) criterion [Rissanen, 1978; Rissanen, 1987] is employed. Description length is defined as the sum of the code length and log-likelihood for a model. Here, both values are calculated in bits.

To examine the effectiveness of the proposed approach, we incorporated the proposed learning mechanism into a model-based diagnostic system. The system accomplishes several tests to narrow down a list of suspected components. In this process, fault probabilities for the suspects are used to select an effective test. The proposed technique is used to presume the probabilities, based on the history of malfunctions for the objective device. The experimental results show that an appropriate model can be derived from a small amount of training data.

The next section describes the inductive learning problem in a probabilistic domain. In Section 3, we introduce the definition of a presumption tree. A criterion for model selection is given in Section 4. A method to utilize the proposed learning mechanism for a performance system is discussed in Section 5. Experimental results are shown in Section 6.

2 Learning in probabilistic domain

In a probabilistic domain, it is assumed that each individual event occurs according to a certain probabilistic distribution. Moreover, it is also assumed that only a few event occurrences can be observed. Therefore, it is necessary to acquire an appropriate probabilistic model based on the observed data, to estimate future events. As an example of such a domain, we consider malfunctions of a device. As shown in Table 2–1, the device is composed of 16 components, where each component has two kinds of attributes, i.e., component type and its age. In this example, malfunctions were observed 32 times.

Table 2–1 Example

Component	Attributes		No. of Obs.
	Type	Age	
1	a	old	1
2	a	new	0
3	b	old	13
4	b	new	9
5	c	old	1
6	c	new	1
7	d	old	0
8	d	new	0
9	e	old	0
10	e	new	0
11	f	old	1
12	f	new	0
13	g	old	0
14	g	new	5
15	h	old	1
16	h	new	0
Total			32

First, we pay attention to the component type. Here, let the fault probability for a component of type **x** be $p(\mathbf{x})$. As shown in Table 2–2, it seems that $p(\mathbf{b})$ is very high and $p(\mathbf{g})$ is also higher than the others. However, it may be dangerous to estimate that "$p(\mathbf{a})$ is higher than $p(\mathbf{d})$" or "$p(\mathbf{c})$ is about twice as large as $p(\mathbf{a})$". The reason is that only a few bits of data are given and it is possible that the observed events happened by chance, while there is little difference among $p(\mathbf{a}), p(\mathbf{c})$ and $p(\mathbf{d})$.

Table 2–2 Fault frequency for each type

Type	No. of Components	No. of Obs.
a	2	1
b	2	22
c	2	2
d	2	0
e	2	0
f	2	1
g	2	5
h	2	1

On the other hand, regarding the age attribute, old components broke down 17 times, and new ones broke down 15 times. Therefore, the information about age is less helpful than the component type to presume the fault probability for each component. However, if the component type is **g**, then the age factor may be important. Hence, in the process of presuming the probabilities of future events, it is important to choose helpful attributes and/or their combinations.

2.1 Presumption problem

This section presents the presumption problem. Consider a set of events $X = \{x_1, x_2, ..., x_m\}$ and attributes $a_1, a_2, ..., a_n$. Here, we assume that the events are exhaustive and mutually exclusive, and that the domain for each attribute a_j $(j = 1, 2, ..., n)$ is a finite set $Dom(a_j)$. As shown in Table 2–3, for each event x_i, a value v_{ij} $(\in Dom(a_j))$ for each attribute a_j is given. Also, n_i, the number of observations is given.

Table 2–3 Table of events

Event	Attributes				No. of Obs. (times)
	a_1	a_2	$\cdots$	a_n	
x_1	v_{11}	v_{12}	$\cdots$	v_{1n}	n_1
x_2	v_{21}	v_{22}	$\cdots$	v_{2n}	n_2
x_3	v_{31}	v_{32}	$\cdots$	v_{3n}	n_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_m	v_{m1}	v_{m2}	$\cdots$	v_{mn}	n_m

The problem is to presume the probability $\hat{p}_i$ for each event x_i $(i = 1, 2, ..., m)$, from the number of observations n_i. However, this task is not easy. Consider two distinct events, x_i and x_j, such that $n_i \neq n_j$. Here, it must be concluded that either $\hat{p}_i = \hat{p}_j$ or $\hat{p}_i \neq \hat{p}_j$. If the number of observations, n_i and n_j, are evidently different, it can be concluded that $\hat{p}_i \neq \hat{p}_j$. However, if there are few differences between n_i and n_j, it may not be concluded that $\hat{p}_i \neq \hat{p}_j$, because the difference may be due to expected random variation.

Here, we consider the two extreme decision strategies as follows.

(a) Only if $n_i = n_j$, conclude that $\hat{p}_i = \hat{p}_j$, otherwise $\hat{p}_i \neq \hat{p}_j$.

(b) Only if n_i and n_j are extremely different, conclude that $\hat{p}_i \neq \hat{p}_j$, otherwise, $\hat{p}_i = \hat{p}_j$.

Although strategy (a) leads to a more precise model, it is very sensitive. That is, it tends to over specialization. On the other hand, strategy (b) is insensitive and tends to over generalization. Consequently, plausible probabilities can not be derived by using these extreme strategies. Moreover, if the number of observed data is small, quite different probabilistic models may become the candidate for the selection. Therefore, a criterion to select the most plausible probabilistic model is necessary.

In the following sections, we introduce a method to resolve such a problem.

3 Presumption tree

In this section, a *presumption tree* is introduced to express a probabilistic model. Using a presumption tree, all the events are classified into several *groups*. Here, each event, x_i, in a group is assumed to have the same probability, $\hat{p}_i$, of occurrence. Therefore, the probabilities for individual events can be derived from a presumption tree. The details are described below.

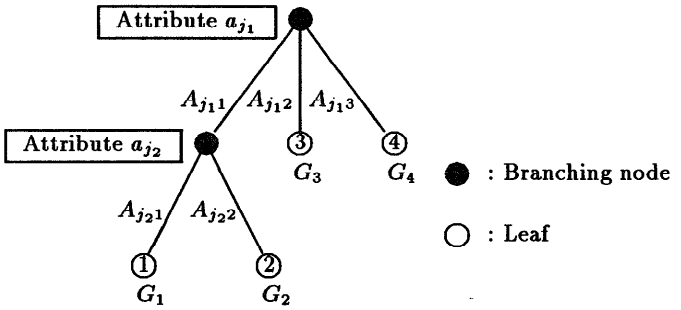

Fig. 3–1 Presumption tree

As shown in Fig. 3–1, a presumption tree consists of several branching nodes and leaves. An attribute a_j corresponds to each branching node, and subset A_{jk} of $Dom(a_j)$ corresponds to each branch from the branching node. Here, each A_{jk} must satisfy the following conditions.

$$A_{jk} \subset Dom(a_j) \qquad \text{(subset)}$$
$$A_{jk} \cap A_{jl} = \phi \quad (k \neq l) \qquad \text{(disjoint)}$$
$$\bigcup_k A_{jk} = Dom(a_j) \qquad \text{(exhaustive)}$$

A presumption tree is used to classify all the events into several groups. For each leaf l, a group G_l of events corresponds to it. For example, for a presumption tree shown in Fig. 3–1, a group G_l of events for each leaf $l\,(l = 1, 2, 3, 4)$ is as follows:

$$G_1 = \{x_i |\ v_{ij_1} \in A_{j_11} \wedge v_{ij_2} \in A_{j_21}\}$$
$$G_2 = \{x_i |\ v_{ij_1} \in A_{j_11} \wedge v_{ij_2} \in A_{j_22}\}$$
$$G_3 = \{x_i |\ v_{ij_1} \in A_{j_12}\}$$
$$G_4 = \{x_i |\ v_{ij_1} \in A_{j_13}\}$$

A presumption tree can be regarded as a description for a probabilistic model by assuming that all the events x_i in a class G_l have the same probability $\hat{p}_i$:

$$\hat{p}_i = \frac{1}{|G_l|} \cdot \frac{O_l}{\sum_k O_k} \qquad (x_i \in G_l)$$

Here, O_l denotes the total number of observations for events in G_l. For example, consider events x_1, x_2, x_3 as shown in Fig. 3–2.

Event	Attribute a_1	No. of Obs.
x_1	X	17
x_2	Y	1
x_3	Y	2

Fig. 3–2 Example

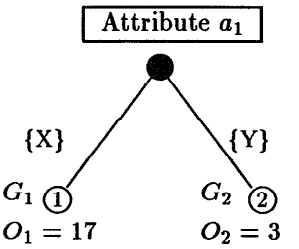

Fig. 3–3 Example presumption tree

Figure 3–3 shows an example of a presumption tree for the events. It indicates that the events are classified into two groups, G_1 and G_2, such that

$$G_1 = \{x_i |\ v_{i1} \in \{X\}\} = \{x_1\}$$
$$G_2 = \{x_i |\ v_{i1} \in \{Y\}\} = \{x_2, x_3\}.$$

The probability $\hat{p}_i$ can be estimated for each event x_i:

$$\hat{p}_1 = \frac{1}{|G_1|} \cdot \frac{|O_1|}{|O_1| + |O_2|} = \frac{17}{20}$$

$$\hat{p}_2 = \hat{p}_3 = \frac{1}{|G_2|} \cdot \frac{|O_2|}{|O_1| + |O_2|} = \frac{3}{40}$$

In the following section, we introduce a criterion for selecting a presumption tree which describes the most appropriate model according to the observed data.

4 Model selection with MDL criterion

As a criterion for the selection, we adopted the minimum description length (MDL) criterion [Rissanen, 1978; Rissanen, 1987; Rissanen, 1986]. He argued that the least description length model is expected to fit for presuming the future events better than any other models. Here, description length for a model is defined as the sum of:

(1) Code length of the model.

(2) Code length of the data w.r.t. the model.

That is, the sum of the model complexity and model fitness for the observed data. The MDL principle is used to induce classification rules, such as a decision tree [Quinlan and Rivest, 1989] or a decision list [Yamanishi, 1989]. In our approach, the MDL criterion is adopted to select the most appropriate presumption tree(i.e., the most plausible probabilistic distribution). We define the description length of a presumption tree as the sum of:

(1) Code length of the tree.

(2) Log-likelihood of data given tree.

The log-likelihood function is frequently used to measure the distance (fitness) of a model and observed data. Here, both of the code length (1), (2) are measured in bits.

Since the calculation for the code length of a tree is very complicated, we restrict the shape of the tree.

Although the selected model may not be the optimal one, it seems near optimal in most cases. The restriction is as follows. For a branching node with l branches, let the corresponding attribute be a_i. Then, A_{ij} $(j = 1, 2, ..., l-1)$ must be a singleton set, and A_{il} is $Dom(a_i) - \bigcup_{j=1}^{l-1} A_{ij}$. Under this restriction, code length(model complexity) L1 for a presumption tree is as follows (see appendix for the proof).

$$L1 = \sum_{x \in P} \{\log(n - d_x) + \log k_x + \log \binom{k_x}{l_x - 1}\}$$

$$+ \sum_{x \in Q} \{\frac{1}{2} \log O_x\} + (|P| + |Q|)$$

Here, P is a set of all the branching nodes and Q is a set of all the leaves. For each branching node x, l_x is the number of branches, d_x is the depth of the node and $k_x = |Dom(a_i)|$ (a_i is the corresponding attribute for node x).

On the other hand, log-likelihood(model fitness) L2 is defined as follows, where, n_i is the number of observations for each event x_i, and $n = \sum n_i$.

$$L2 = -\sum_{i=1}^{m} n_i \, (\log \hat{p}_i - \log \frac{n_i}{n})$$

An example of the calculation is given below. Fig. 4–1 shows three example presumption trees for the device malfunction example in section 2. Tree A is a trivial one. It estimates the probabilities of all the events being equal, i.e., $\hat{p}_i = 1/16$ $(i = 1, 2, ..., 16)$. Tree B classifies events into 4 groups(i.e. type **b**, type g & old, type g & new and others), and tree C classifies events into 16 groups.

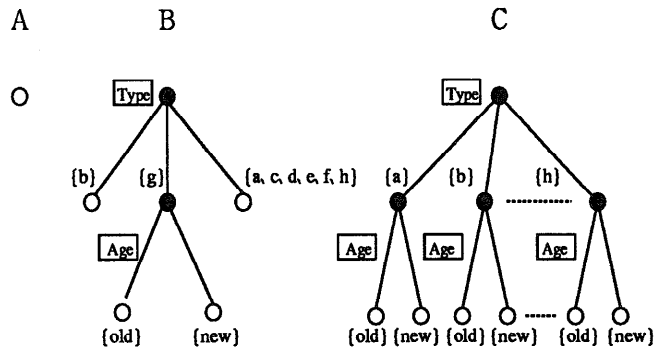

Fig. 4–1 Example of presumption trees

The description length for each tree is shown in Table 4–1. Model A has the shortest code length(L1), but its data description length(L2) is large. On the other hand, model C is just the opposite. Model B has the least description length(L1+L2). Therefore, utilizing the MDL criterion, model B is the most appropriate one among the three. This result agrees with our intuition.

Table 4–1 Individual description length

Model	L1	L2	Total Length L1+L2 (bits)
A	3.5	56.2	59.7
B	16.2	6.8	23.0
C	49.6	0.0	49.6

5 Application to performance systems

An induced probabilistic model can be used to improve the efficiency of certain kinds of performance systems. In several performance systems, the expected computation costs can be estimated by using information about the probabilistic distribution of events. Therefore, if the most appropriate probabilistic model is derived, it is possible to select a computation strategy with the minimum expected cost.

For example, by using a probabilistic distribution, a model-based diagnostic system can estimate the expected information gain for each possible test, and can select the most appropriate one[de Kleer and Williams, 1987; de Kleer and Williams, 1989]. de Kleer and Williams introduced the *minimum entropy* technique where entropy is calculated from the fault probability for each suspected component and is used to evaluate a test to be carried out next. That is, the system calculates the expected entropy gain(information gain) for each possible test, and selects the most appropriate one.

However, if the presumed fault probability distribution is quite different from the real one, the calculation for the expected information gain is meaningless. Therefore, we must acquire a precise probabilistic distribution. For example, consider a communication system, which consists of 100 modems $(m_1, m_2, ..., m_{100})$, 100 terminals$(t_1, t_2, ..., t_{100})$ and a bus(b_1). Suppose malfunctions in the system were observed 10 times and all of the faulty components were distinct modems, say $m_{i_1}, m_{i_2}, ..., m_{i_{10}}$. By a simple estimation, the each fault probability for $m_{i_1}, m_{i_2}, ..., m_{i_{10}}$ is 1/10, and that for the other 191 components are all 0. However, intuitively, it is natural to estimate that all the modems have higher fault probabilities than the other components. By using the proposed technique, such a model can be derived. The difference between the two estimations mentioned above is considered to affect the performance of the diagnosis. The details of the experimental results are shown in the next section.

Another application is to utilize deductively acquired knowledge. Although a deductive learning mechanism, such as EBL[Mitchell and Keller, 1986; DeJong and Mooney, 1986] or chunking [Rosenbloom and Newell, 1982] can be used to acquire knowledge, it does not always improve the system performance. A strategy to use the acquired knowledge greatly affects the performance[Minton, 1988; Greiner and Likuski, 1989]. Therefore, to acquire an appropriate strategy, it is indispensable to presume future events based on experience. Our approach is considered to be effective for

such applications.

6 Experimental results

To examine the effectiveness of the proposed approach, the learning mechanism was incorporated into a model-based diagnostic system. The system performs the following procedures repeatedly until a faulty component is pinpointed.

1. Analyzes the symptom and/or the test results by using the model-based knowledge, and then creates a list of suspected components.
2. Selects and performs the most appropriate test to narrow down the suspected components.

In step 2, estimation of the effectiveness of each possible test(i.e., the expected information gain) is calculated by using the fault probability for each suspected component.

In the experiments, the fault probability for each component is estimated two ways, i.e., the proposed method and a simple estimation method. For each derived probabilistic distribution, the average system performance is examined. The details are described below.

The objective is to diagnose a communication system as discussed in the previous section. Here, assume that the components are classified into three groups as shown in Table 6–1. We also assume that the fault probabilities for group 1, 2, 3 are 0.33, 0.66, 0.01, respectively, and each component in a group has the same probability(e.g., $p(m_1) = p(m_2) = \cdots = p(m_{50}) = 0.66/50 = 0.0132$).

Table 6–1 Model of communication system malfunction

Group	Components	Type	Age
1	$m_1, m_2, ..., m_{50}$	modem	old
2	$m_{51}, m_{52}, ..., m_{100}$	modem	new
3	$t_1, t_2, ..., t_{100}$	terminal	new
	b_1	bus	new

At first, several faults are artificially generated as training examples, according to the probabilistic distribution as shown above. From these examples, the most appropriate probabilistic model (presumption tree) is derived by the proposed mechanism. By using the derived model, the fault probability for each component is presumed. On the other hand, to make comparisons, we estimated each probability in a simple manner by assuming the probabilities to be proportional to the number of observations.

In order to compare these two estimated probabilistic distributions, additional 100 faults are generated, according to the probabilistic distribution. The average numbers of required tests were compared and the results are shown in Fig. 6–1. The model that is derived by the proposed mechanism could classify the events into three correct groups based on only 20 training data examples. Therefore, the system performance could be improved according to such a small amount of training data, while a simple estimation method requires a great amount of training data to gain an equivalent performance level.

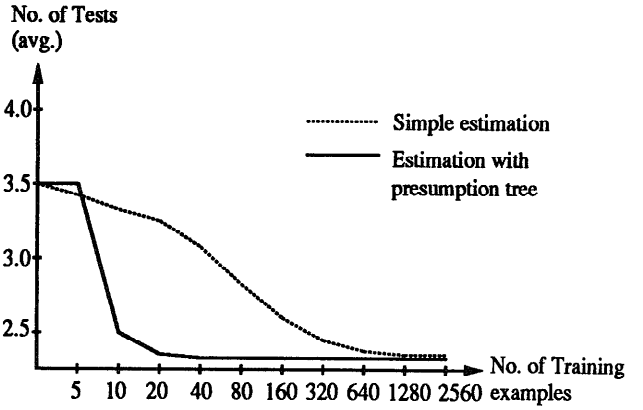

Fig. 6–1 The effect of learning

7 Conclusion

An inductive learning method in probabilistic domain is introduced. A presumption tree is adopted to describe a probabilistic model. It is shown that the most plausible model can be derived from a small amount of observation data by using the proposed technique. Also, it is shown that the derived model can be used to presume the probability distribution based on the experience, and can control the execution of a performance system to improve its efficiency.

Although the proposed mechanism works well, it searches all possible presumption trees to derive the least description length tree. Hence, with the growth in the number of attributes, much computation time would be required. Therefore, it is necessary to develop an algorithm with a more sophisticated searching strategy, such as 'branch and bound' technique.

Acknowledgment

The authors would like to express their thanks to Tatsuo Ishiguro, Yoshihiro Nagai and Tomoyuki Fujita for their encouragement in this work. Further, they also thank Kenji Ikoma of the Institute for New Generation Computer Technology.

Appendix

The code length for a presumption tree is calculated in manner similar to that reported by [Quinlan and Rivest, 1989]. Here, we assume that $-\log p$ bits are required to describe information with probability p. The code length is calculated as the sum of individual code lengths for the following information.

(1) Category of each node.

(2) Corresponding attribute for each branching node.

(3) Corresponding value set for each branch.

(4) Estimated probability for each leaf.

To describe the information about (1), it requires 1 bit/node, because the category for a node could be a branching node or leaf. Hence, $|P| + |Q|$ bits are required in total. The information about (2) requires $\sum_{x \in P} \log(n - d_x)$ bits, because the attribute is one of the $n - d_x$ attributes. Next, we calculate the code length for (3). The number of branch l_x for a node x can be described in $\log k_x$ bits. A_{ij} $(j = 1, 2, ..., l_x - 1)$ is a singleton subset of $Dom(a_i)$, and we do not care about their order, therefore the description for A_{ij} requires $\log \left(\begin{matrix} k_x \\ l_x - 1 \end{matrix} \right)$ bits. Hence, total code length for (3) is:

$$\sum_{x \in P} \{\log k_x + \log \left(\begin{matrix} k_x \\ l_x - 1 \end{matrix} \right)\}$$

Finally, the code length for (4) is $\sum_{x \in Q} \{\frac{1}{2} \log O_x\}$ ([Yamanishi, 1989]). Consequently, total code length is:

$$\sum_{x \in P} \{\log(n - d_x) + \log k_x + \log \left(\begin{matrix} k_x \\ l_x - 1 \end{matrix} \right)\}$$

$$+ \sum_{x \in Q} \{\frac{1}{2} \log O_x\} + (|P| + |Q|)$$

References

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[de Kleer and Williams, 1989] J. de Kleer and B. C. Williams. Diagnosis with behavioral modes. *Proc. IJCAI-89*, 2:1324–1330, 1989.

[DeJong and Mooney, 1986] G. F. DeJong and R. Mooney. Explanation-based learning: An alternative view. *Machine Learning*, 1(2):145–176, 1986.

[Greiner and Likuski, 1989] R. Greiner and J. Likuski. Incorporating redundant learned rules: A preliminary formal analysis of EBL. *Proc. IJCAI-89*, 1:744–749, 1989.

[Koseki, 1989] Y. Koseki. Experience learning in model-based diagnostic systems. *Proc. IJCAI-89*, 2:1356–1361, 1989.

[Mingers, 1989] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4:227–243, 1989.

[Minton, 1988] S. Minton. Quantitative results concerning the utility of explanation-Based learning. *Proc. AAAI-88*, 2:564–569, 1988.

[Mitchell and Keller, 1986] T. M. Mitchell and R. M. Keller. Explanation-Based generalization: A unifying view. *Machine Learning*, 1(1):47–80, 1986.

[Quinlan and Rivest, 1989] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80(3):227–248, 1989.

[Quinlan, 1986] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[Rissanen, 1986] J. Rissanen. Complexity of stings in the class of Markov sources. *IEEE Trans. on Information Theory*, 32(4):526–531, 1986.

[Rissanen, 1987] J. Rissanen. Stochastic complexity. *Jnl. Roy. statist. Soc. B*, 49(3):223–239, 1987.

[Rosenbloom and Newell, 1982] P. S. Rosenbloom and A. Newell. Learning by chunking summary of a task and a model. *Proc. AAAI-82*, pages 255–257, 1982.

[Yamanishi, 1989] K. Yamanishi. Inductive inference and learning criterion of stochastic classification rules with hierarchical parameter structures. *Proc. SITA-89*, 1989.