Finding and Using Multiple Models For Autonomous Learning

I.N.P. Davidson
CSIRO - Mathematical and Information Sciences
723 Swanston St, Carlton, Victoria, Australia 3053

ABSTRACT

Achieving autonomous learning systems which can govern themselves is one of the goals of A.I. Most learning systems explore a fixed model space to explain a set of data. We believe that the "best" but most distinct models in the available space can provide insight into questions of autonomy such as when to change the model space and how to generate new data points (via experiments). We explore this idea by focusing on clustering problems where the initial data is known to be insufficient to find the true model. We propose a method to generate new data points via experiments. Our approach results in convergence to the true model using half as many additional data points than if they were randomly selected.

KEYWORDS: *Autonomous learning, machine discovery, clustering, unsupervised learning*

1. INTRODUCTION AND MOTIVATION

If inductive learning aims at answering the question, "What does the data tell us ?", autonomous learning adds the question, "What can we now do to better understand the domain ?".

So what can we do to better understand the domain to which we are applying our learning system ? Inductive learning like most artificial intelligence problems is inherently a search through a predefined model space. Most inductive learning tools whether they be unsupervised [1] or supervised [2], *primarily* focus on finding the single best model with respect to some criterion for a fixed set of data. This is quite adequate if the tool is to be used by a human who can interpret the results and make appropriate changes. To make such tools autonomously learn more about a domain we must address problems of how to change the model space and how to generate new data. It is our belief that finding and using multiple models can provide insight into these more complex questions associated with autonomous learning.

In this paper we focus on using multiple models to answer the question, "Given the current data and the best model(s) found, what should be the next set of experiments to conduct be to find the true model for the domain ?". Which model is better for a given set of data has been addressed by the minimal encoding length approach independently proposed by Wallace (1968) [1] and Rissanen (1978) [3]. Their approach has the benefit that complex models are chosen over simpler ones only if the data available justifies it. But to our knowledge the approach provides no indication of how to generate new data points. Whilst we focus on this question in this discourse, we believe our approach could be used to determine how to change the model space and other questions associated with autonomous learning. We intend to explore these at a latter time.

This paper documents our approach for finding and using multiple models for clustering problems otherwise known as unsupervised learning. The paper is divided into a further six sections. The first is a basic introduction to clustering which provides the terminology used throughout this paper. In the next section we define in limited detail the clustering system we have developed (a more complete description exists [4]). The criterion used to evaluate each model (the minimum message length) and our search mechanism (simulated annealing) are described. The subsequent sections outline how we search the model space to find multiple models and then how these can be used to answer our next experiment question. The final two sections discuss and conclude our current work and touches on future research.

2. AN INTRODUCTION TO CLUSTERING

Clustering, also called unsupervised or intrinsic classification, has a long history in numerical taxonomy [5] and machine learning [6]. Clustering attempts to find groups within data so as to better understand the domain the data is from. It has been applied to generation of taxonomies for flora and fauna, concept formation and data mining. The objects/entities to be clustered are each described by a set of *d* attributes. Clustering involves determining the number of classes (groups), a description for each class which can be used to determine membership and assigning each object to one or more of these classes. As the number of classes is unknown and no pre-classified training set exists, clustering is unsupervised. The collection of classes and their descriptions form a taxonomy/model of the objects.

A clustering system contains three major parts. The *knowledge representation scheme* (KRS) which defines the searchable model space. The *criterion* which provides a "goodness" measure for each model and the *search mechanism* which explores the model space attempting to find the model which leads to the optimal criterion value.

The KRS determines the type of classes and their possible interrelationships. A dichotomy for clustering options which impact on the KRS has been defined elsewhere [7]. The criterion evaluates the "goodness" of each of the models. It is usually a real value function that takes as parameters the objects and/or class descriptions and is the objective function of the search.

The search mechanism explores the model space attempting to find the best model by finding the optimal (either minimum or maximum) value of the objective function. For all but the most restrictive model spaces the number of possible models to evaluate is combinatorially large. Exhaustively evaluating each model is not even considered as a search mechanism. The search mechanism must consistently find the global optima or at least a good local optima in a number of different application domains with a minimum of computation.

3. OUR CLUSTERING SYSTEM

The clustering system developed merges together two problem-invariant (robust) technologies: the minimum message length criterion (MML) and simulated annealing (SA). This has so far shown to result in a clustering system which can be applied to a number of different problems with minimum changes. The objective function of our search is to minimize the message length for non-hierarchical and probabilistic classes which objects are exclusively assigned to. However, most large and interesting search problems possess many local optima [14]. We feel that SA is a good search mechanism to explore these complex model spaces, since it can escape local minima [14]. In the following sub-sections we describe the two technologies.

3.1 The Minimum Message Length Criterion

Chaitin [8], Kolmogorov [9] and Solmonoff [10] in varying forms independently proposed algorithmic information theory (AIT). AIT intuitively allows us to quantify the notion of complexity and compressibility of objects. Learning by induction is inherently about compressing observations (the objects) into a theory (the model). Boyle's law ($P = k.N/V$) on ideal gases relates the number of molecules (N) in a measurable closed volume (V) to pressure (P). A table could store every possible combination of N and V and the resultant pressure. However, Boyle's law compresses this table into a much shorter description, the above equation.

Wallace and Boulton [1], extend this compressibility notion into their minimum message length (MML) approach to induction. They define a criterion which can be used to select the most probable model from a given set of mutually exclusive and exhaustive models, H^*, for the objects, D. The MML approach specifies that the minimal encoding of the model and the objects given the model is the best. In terms of Bayes theorem, we wish to maximise the posterior distribution, $P(H_i \mid D,c)$ where c is the background context:

$$P(H_i|D,c) = \frac{P(H_i|c).P(D|H_i,c)}{P(D)} \tag{1}$$

Taking the logarithm of this expression yields

$$-\log P(H_i|D,c) = -\log P(H_i|c) + -\log P(D|H_i,c) + const \tag{2}$$

Our interest is in comparing relative probabilities so we can ignore *const.* Information theory [Shannon] tells us that - log *(P(occurrence))* is the minimum length in bits to encode the occurrence. Hence by minimising equation (2) we inherently maximise the posterior distribution and find the most probable model. The expression to minimise has two parts, the first being the encoding of the model and the second the encoding of the objects given the model. The object collection is random if the size of encoding the model and the objects given the model is approximately equal to the size of directly encoding the objects. That is there is no way to compress the objects into a shorter description/theory. The two part message is precisely described for intrinsic non-hierarchical classification [1] and [11].

The MML criterion only defines a goodness measure for a model with an inherent bias towards simple models. It does not indicate how to search the model space. To do that we use simulated annealing.

3.2 Searching The Model Space Using Simulated Annealing

The Metropolis criterion was first used as a Monte Carlo method for the evaluation of state equations in statistical mechanics by Metropolis et al. [12]. Kirkpatrick et al. [13] demonstrated how using the Metropolis criterion as a test in iterative optimisation can solve large combinatorial optimisation problems. They called their approach the simulated annealing technique as it mimics the annealing of a piece of metal to minimise the energy state of the molecules within the metal. SA is an iterative local optimisation technique. At any time there is only one current solution which is slightly changed at each iteration. As SA is a Markov process the current solution, S_n, at time n, is a result of the perturbation of solution S_{n-1}. The algorithm continually perturbs the current solution to generate new candidate solutions. SA unconditionally accepts candidates of better quality than the previous solution and conditionally accepts those of a worse quality with a probability p, where:

$$p = e^{-\left(\frac{|Difference\ in\ Quality|}{System\ Temperature}\right)} \tag{3}$$

Worse quality solutions can be accepted which allows the search to escape from local minima which are common in most complex search problems [13]. We set the initial temperature T_0, so there is a 90% probability of accepting an increase in cost. This probability decreases as the temperature decreases. The cooling constant, R reduces the temperature such that, $T_k = T_{k-1}.R$.

$$T_0 = -\frac{C_0}{\log_e(0.9)} \tag{4}$$

C_0 is the goodness evaluation of the initial solution.

The implementation of our algorithm can be found in [4].

Simulated annealing statistically guarantees that the global optimum will be found, if the thermodynamic equilibrium is reached at every temperature and the cooling schedule is slow enough [14]. However, this is an infinitely long process. We do not maintain these two requirements due to the need to find a solution in finite time. Instead, after a fixed number of iterations at a temperature, the temperature is reduced and the cooling constant provides discrete changes in the temperature. However non-ideal SA approaches, such as the one we use, still find good, local optima solutions [14].

4. FINDING MULTIPLE MODELS

Our thesis is that distinct but good models can be used to generate new experiments whose results can be used to better understand the domain. This requires finding the *n* models which provide the best values for the objective function *but are sufficiently different from each other*. Just finding the *n* best models would most likely result in finding a good model and slight variations of it.

To achieve our aim we must handle two key issues. Firstly, we must be able to quantify the difference between two models. Secondly we must adjust our search mechanism. Let us discuss the first.

4.1 Quantifying The Difference Between Models

A model can be characterised by its predictions or its syntactic description. A model (the taxonomy) makes predictions on how to group together objects. Each model assigns each object to a cluster. For two clusters from different models, we can measure the similarity between them by counting the number of common objects. For two models we can measure their similarity by counting the number of common objects for every possible combination of cluster pairs (one from each model). This is inherently a measure of the common "cluster neighbourhood" (clusterhood) each entity has in two different models. This measure can be achieved by building a $r \times c$ contingency table, *P*, where *r* is the number of clusters in model A (M_A) and *c* is the number of clusters in model B (M_B). The cell P_{ij} in the table holds the number of objects common to cluster *i* in M_A and cluster *j* in M_B. The total count of the table will be the number of objects/entities we are clustering. Where M_A is the same as M_B only the leading diagonal of the resultant table will contain non-zero elements.

A model can also be characterised by its description. In clustering, a model consists of classes and their descriptions. In our approach each class description contains a probability distribution for each attribute. The message we construct (whose length we are trying to minimise) only encodes an attribute distribution of a class if it is sufficiently different from the population's (collection of all objects) distribution for that attribute. We can characterise the descriptive difference between two models in a contingency table, *D*, which has the same structure as the contingency table *P*. For each attribute we can determine which of the clusters for each model has a distribution for that attribute that is the greatest from the populations. The cell D_{ij} holds the count of attributes for cluster *i* in M_A and cluster *j* in M_B whose probability distribution is of greatest distance from the population's distribution. The total count for the table is the number of attributes. The contigency table inherently holds the distinguishing features (attributes) of each class.

The contingency tables *P* and *D* contain the differences between the two models *A* and *B* in their most rudimentary forms (predictions and descriptions). We can use this information to measure if a relationship exists between the two models. Note we do not attempt to determine what the relationship is, only if it exists. This can be achieved by using a number of different contingency table association measures [15]. We choose the Goodman and Kruskal lambda measure of predictive ability because it is both a readily interpretable probability measure and is not symmetrical. The measure λ_{AB} measures the ability to predict the cluster in model *B* given we know the cluster in model *A*. It should be noted that $\lambda_{AB} \neq \lambda_{BA}$ is generally true. That is, *A* may be predictable from *B* but not *B* from *A* and vice versa. Specifically λ_{AB} calculates the relative decrease in the probability of an error in guessing the class given by model *A* if the class for model *B* is known. Formally we can write:

$$\lambda_{AB} = \frac{P_1 - P_2}{P_1}$$
$$P_1 = 1 - \frac{\max(n_{.i})}{N} \tag{5}$$
$$P_2 = \sum_{i=1}^{r}\left(1 - \left[\frac{\max(n_{i1} \dots n_{iC})}{n_{i.}}\right]\right) . \frac{\max(n_{i1} \dots n_{iC})}{N}$$

where $n_{.i}$ is the total of column *i*, $n_{j.}$ is the total of row *j* and n_{ij} is the value of the contingency table at row *i* and column *j*.

By calculating the lambda value for the tables, *P* ($\lambda_{AB}(P)$) and *D* ($\lambda_{AB}(D)$) we can measure the predictability of a model from another in terms of predictions and descriptions respectively.

4.2 Adjusting The Search

The ideal annealing algorithm converges to the global optimum. However the trajectory through the model space in getting there may not be sufficiently diverse to find other good but different local optima. We must therefore adjust our search method to be consistent with our aim. We can achieve this by introducing a bias which guides the search away from already found good local optima. This is facilitated by storing *n* models which are the best (with respect to the objective function) but sufficiently different from each other. These models are the best and most diverse models known.

Models are only considered to be stored if their message length is less than any of the currently stored models. To be stored, the summation of the models predictability from every other stored model must be less than this same measure for one of the currently stored models. The model whose predictability is the greatest is replaced. Predictability is calculated using the λ_{AB} measures for either the *P* or *D* contingency tables. Candidate models have a penalty added to their "goodness" value in proportion to their similarity to the stored models.

5. THE USES OF MULTIPLE MODELS

In the previous section we defined two measures of difference between models. These measures can be encoded in a contingency table and the predictability of one model from another calculated. How we should use these measures to influence our search depends on what we are trying to achieve. In this paper we focus on what the next best set of experiments to conduct are.

The question of how to guide the next experiments to conduct has been addressed in Lenat's work on AM [16] and Kulkarni and Simon's work on Kekada [6]. Lenat described the notion of "interestingness" and felt the system should focus its attention on interesting phenomena. Similarly, Kekada focuses its next experiments on surprising phenomena, believing that if a result of an experiment was unexpected then the knowledge of that area of the domain is obviously lacking and should be explored. Both approaches use heuristics to describe the notions of interestingness and surprise. As we hope to have available the best but most different models we focus the next set of experiments where these models' predictions differ. By doing this we can resolve which of these models is the better for the domain. By continually running the clustering system, finding distinct but good models, and then generating data points where these models' predictions differ we generate data points where our knowledge of the domain is contradictory.

To determine which of measure of predictability (model description or predictability) is better we conducted experiments on the following problem. Consider a population of objects/entities each having *m* binary attributes. In the population there exists *m* classes. Class *i*, *i* = 1 .. *m* can be precisely described as having the value 0 (false) for all attribute except the *ith* which is 1 (true). Table 1 provides the precise description for a few classes for the *m*=10 situation.

Attribute	1	2	3	4	5	6	7	8	9	10
Class 1	1	0	0	0	0	0	0	0	0	0
Class 10	0	0	0	0	0	0	0	0	0	1

Table 1. Precise description for classes in m=10 situation.

To determine the proportion of each class in the population we make use that the summation of the first *r* integers is given by:

$$S_r = \frac{r.(r+1)}{2}$$

(6)

from this the relative proportion, $P_{i,\, i\, =\, 1\ldots\, m}$ of class *i* in the population is given by:

$$P_i = \frac{2i}{m(m+1)}, \qquad \sum_{i=1}^{m} P_i = 1$$

(7)

By using the MML equations described in [1] we can determine the approximate number of objects (data points) required to find the true model if the objects are randomly sampled from the population. For our trial set of objects this number was 184. Below this amount of data the best model is to place all objects into one class, indicating that the data from an information theoretic view is random. For this data set the encoding of the model and data for the true model and one class model were 566.21 and 570.24 nits respectively. The difference between the lengths is the comparitive difference in likelihood. Thus the true model is approximately e^4 times more likely than the one class model for the given data set.

We conducted trials to determine how successful our strategy of focusing experiments on where the models predictions differ is. In each trial the clusterer was given the first 60, 80 and 120 objects of our data set. As we have shown, this is insufficient to chose the true model over the one class model. For 120 objects the true model and one class model had encoding length of 415.56 and 368.32 nits respectively for this reduced data set. The true model is approximately e^{47} times less likely than the one class model for the data. The class distributions follow in table 2:

Class Number	1	2	3	4	5	6	7	8	9	10
Frequency	5	1	7	13	7	13	15	17	21	21

Table 2. Class distributions for initial 120 objects

Our aim is to generate new data points so that eventually the true model is found. The number of new experiments required to converge to the true model is one obvious measure of performance.

5.1 How To Generate New Experimental Data

The process of running the clustering system with a given set of data produces a number of theories (taxonomies/models) of the data. We wish to generate new experimental data which can be used in further applications of the clustering system to better understand the domain and find the true model. We focus on generating new data where the predictions of the theories are different. An example based approach is used where an example of an object the models' predictions disagree upon is used as input into an experiment. The experiment takes the object as an input and returns similar objects. For additional complexity there is a stochastic aspect to the experiments which results in the chance that the experiment will return the wrong result. In our studies this error is 25%.

The examples can be selected by re-arranging a *P* contingency table so that the leading diagonal has the largest counts. The remaining elements represent objects which are predicted indifferently for these two models. Completing this task for all possible pairs of models can determine those objects for which the model's predictions differ the greatest.

Experiments could also be generated by prescription. This would involve a description of an exemplar object for which, if it were to exist, the current stored models would make contradictory predictions for. This exemplar could be constructed from where cluster descriptions differ the most between all clusters from one model with all clusters from another. We have not explored this option as yet.

We established two control trials. One generated new objects by sampling them from the population (sampPop) whilst another generated new objects from each class in equal proportion (equProp).

Table 3 illustrates the comparison between each model search and experiment generation technique. Both search techniques stored the five best models which had the shortest message lengths but were different from each other. The techniques differed in the notion of difference. Search technique A used the predictive difference between models; B the description difference between models. Two experiment generation techniques were tried: technique C generates new objects in batches of 10 whilst technique D generated objects in batches of 20. After each batch was generated, the clusterer was re-run and the process repeated until the true model was discovered. The control trial generated objects in batches of 5. All four of our variations outperformed the two control approaches by requiring approximately half as many data points to converge to the true model.

Search Technique	A	B	A	B	samp	equ
Experiment Generation Technique	C	C	D	D	Pop	Prop
New objects required to find true model. 60 initial objects.	70	70	80	80	170	140
Additional objects required to converge to true model. 80 initial objects.	50	60	80	60	140	120
Additional objects required to converge to true model. 120 initial objects.	40	40	60	60	100	80

Table 3. Results of trials

6. DISCUSSION AND FUTURE WORK

We shall focus our discussion on the situation with 120 initial objects using search technique A (model difference measured by predictions) and experiment generation technique C (batch sizes of 10). The system behavior is summarized in table 4.

	1st Trial	**2nd Trial**	**3rd Trial**	**4th Trial**	**5th Trial**
True Classes Found In One of the Best Models	6	8	7	9	10
Class experiments focus on	1	3	9	2	
True Model Found	No	No	No	No	Yes

Table 4. Summary of Behavior For 120 Initial Object Case.

The ten classes in the true model are not justified by the initial data. After the first trial with 120 objects, the best models, *in combination*, contained the correct description and object assignments for six of these classes. The models most disagreed upon what class objects in the class 1 should belonged to, more objects similar to this class were requested. A further 4 more trials occurred before the true model was found. Of course in our situation we know what the true model is. It was interesting that the number of true classes found did not increase monotonically, nor that the class the models predictions differed most on, was not the least frequent.

One of our aims was to determine the impact of searching the model space to find the best but most different models with respect to description and predictions. However irregardless of the measure of difference used, similar models were found. We feel this is due to the simplicity of the problem and most likely this will not occur in more complex domains.

We have not made use if there exists any relationship between the similarity of two models for their predictive capability and descriptions. We can consider five cases:

Relationship	*Interpretation*
$\lambda_{AB}(P) > \lambda_{AB}(D)$	The models predictions are more similar than their descriptions.
$\lambda_{AB}(P) >> \lambda_{AB}(D)$	The models predictions are significantly more similar than their descriptions.
$\lambda_{AB}(P) < \lambda_{AB}(D)$	The models descriptions are more similar than their predictions.
$\lambda_{AB}(P) << \lambda_{AB}(D)$	The models descriptions are significantly more similar than their predictions.
$\lambda_{AB}(P) \approx \lambda_{AB}(D)$	The similarity between the models with respect to their descriptions and predictions are fairly equivalent.

Table 5. The relationship between measures of similarity between a models description and its predictions.

We can diagramatically represent each situation by considering a population of things which only has one attribute, which we believe to be normally distributed. Each model has only one class whose description is the mean and standard deviation for that particular attribute. Figure 1 illustrate the situations for $\lambda_{AB}(P) > \lambda_{AB}(D)$. The circles represented data points. In this situation the models make similar predictions for the current data points, but they are evidently different. $\lambda_{AB}(P)$ is larger than $\lambda_{AB}(D)$ because the current data points do not occur in areas where the models predictions would differ. Using and contrasting both measures could be of benefit.

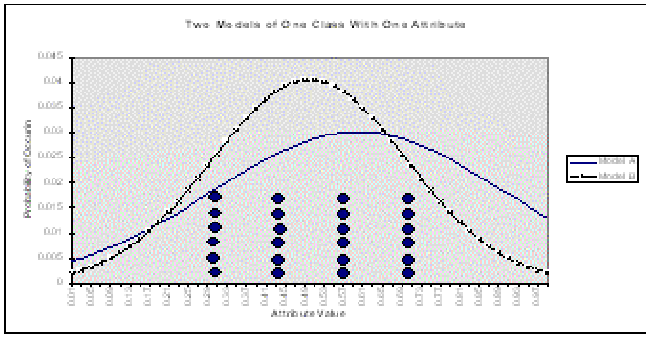

Figure 1. $\lambda_{AB}(P) > \lambda_{AB}(D)$

We intend to explore the annealing literature to see if any insight can be provided to bias the search technique to better explore the model space. The lambda measures of association used in the contingency tables whilst adequate are problematic for skewed distributions. We intend to explore measuring the information content of contingency tables to obtain better measures of predictability. As stated earlier we believe that using multiple models can address other questions in autonomous learning such as when to change the model space which we plan to explore. Our clustering system can change the model space by taking the Cartesian product of attributes and changing the probability distribution (discrete or normal) assumed for each attribute.

7. CONCLUSION

We have developed a clustering system which can search the model space for good but distinct models. The difference between models can be measured regarding their predictions or descriptions. Using these models can provide insight into how to address questions of autonomous learning systems of which, we have focused on the next set of experiments to conduct to better understand the domain. We have applied this clustering system to an artificial problem where the initial set of data is inadequate to find the true model. We explore the idea of generating new objects where the models predictions differ. We have shown that this approach results in finding the true model by generating only half as many additional objects than by using blind techniques for our problem.

8. ACKNOWLEDGMENTS

I am most grateful to Emeritus Professor C.S. Wallace from Monash University for his continual comment and suggestions on this work. Professor Wallace's work is supported by ARC grants A49703162 and A49703170.

9. REFERENCES

[1] Wallace C, Boulton D., "An Information Measure for Classification," *Computer Journal*, Volume 11 No. 2, pp. 185-194, 1968.

[2] Quinlan J.R. "Induction of Decision Trees," *Machine Learning*, Volume 1, pp. 81-106, 1986.

[3] Rissanen J. "Modeling by the shortest data description," *Automatica*, Volume 14, pp. 465-471, 1978.

[4] Davidson I.N.P. "Clustering Using The Minimum Message Length Criterion and Simulated Annealing," in Proceedings of the 3rd International A.I. Workshop, Brno, Czech Republic 1996.

[5] Dunn G., Everitt B.S.. *An Introduction to Mathematical Taxonomy*. Cambridge University Press 1982.

[6] Michalski R.S., Stepp R.. "Learning from Observation: Conceptual Clustering," in Michalski R.S., Carbonell J.G, Mitchell T.M. editors, *Machine Learning: An Artificial Intelligence Approach*, CA: Morgan Kaufmann, 1983.

[7] Clifford T., Stephenson W. *An Introduction To Numerical Classification*, Academic Press, 1975.

[8] Chaitin G. "On The Difficulty of Computations," *IEEE Transactions of Information Theory*, IT-16, 1970, pp. 5-9.

[9] Kolmogorov A. "Logical Basis for Information Theory and Probability Theory," *IEEE Transactions of Information Theory and Control*, IT-14, 1965, pp. 662-664.

[10] Solomonoff R. "A Formal Theory of Inductive Inference: Part 1," *IEEE Transactions of Information Theory and Control*, IT:7, 1964, pp. 1-22.

[11] Wallace C.S., "An Improved Program for Classification," in Proceedings of the 9th Australian Computer Science Conference, Volume 8 No. 1, 1986, pp. 357-366.

[12] Metropolis N., et al, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, Volume 21 #6 June, 1953, pp. 1087-1092.

[13] Kirkpatrick S., Gelati C., Vecchi M. "Optimization by Simulated Annealing," *Science*, Volume 220, 1983, pp. 671-680.

[14] Van Laarhoven L. *Theoretical and Computational Aspects of Simulated Annealing*, CWI Tract, 1988.

[15] Everitt B. *Contingency Tables*, Cambridge University Press, 1980.

[16] Lenat D. "AM: An artificial intelligence approach to discovery in mathematics as heuristic search," *In Knowledge-based systems in A.I.*, McGrawHill, 1982.

[17] Kulkarni D., Simon H.A., *Computational Models of Scientific Discovery and Theory Formation*, Editors P. Langely, and J. Shrager, chapter 9, Morgan Kaufmann, 1990.