Reports

Machine Learning and Inference Laboratory

The Distribution Approximation Approach to Learning from Aggregated Data

Ryszard S. Michalski and Janusz Wojtusiak

MLI 08-2

May 30, 2008

George Mason University

THE DISTRIBUTION APPROXIMATION APPROACH TO LEARNING FROM AGGREGATED DATA

Ryszard S. Michalski[1,3] and Janusz Wojtusiak[1,2]

[1] Machine Learning and Inference Laboratory
[2] Department of Health Administration and Policy
College of Health and Human Services
George Mason University
Fairfax, VA 22030, USA

[3] Institute of Computer Science,
Polish Academy of Sciences
Warsaw, Poland

Earlier version of this report has been prepared by Dr. Ryszard S. Michalski before his death in September 2007. The paper has been finished and extended by the second author.

Abstract

This report describes four very simple methods for preparing training data sets for machine learning when training examples are split to several subsets, and only aggregated values of attributes are available for each subset. Specifically, each subset of examples is represented by a vector of pairs of values for each attribute. The first value in the pair is the mean, and the second is the standard deviation of the attribute. Presented methods, M0, M1, M2, and M3, employ consecutively more accurate approximations of the normal distribution of values for each attribute. The consecutive approximations require, however, increasingly larger sets of training examples for each subset in the class.

Keywords: Machine Learning, Aggregated Data, AQ learning, Natural Induction, Computational Learning.

Acknowledgments

The authors thank Professor James Gentle for explaining how to use the software package R to generate samples following normal distribution in method M3, and for suggesting the size of the samples to be generated. Jarek Pietrzykowski helped in preparing the data and the figures in R, preprocessing the data, and providing valuable comments that helped to improve this paper.

This research has been conducted in the Machine Learning and Inference Laboratory of George Mason University, whose research activities have been supported in part by the National Science Foundation Grants No. IIS 9906858 and IIS 0097476, and in part by the UMBC/LUCITE #32 grant. The findings and opinions expressed here are those of the authors, and do not necessarily reflect those of the above sponsoring organization.

1 Introduction

In some application domains training data for concept learning is available not in the standard form of training instances, but in an aggregated form. For example, in medicine, due to the privacy protection, data may be available only in the form of means and standard deviations of values of individual medical parameters for groups of patients. Such data are common in medical literature. Typical methods for machine learning are, however, oriented toward learning from training sets of specific instances, not from aggregated data, hence a problem arises as to how to adopt existing methods to learn from such training sets.

This paper presents an approach to learning from aggregated data that develops an approximation of the distribution of attribute values, and uses them to create training sets for learning. Based on this idea, four simple methods, M0, M1, M2, and M3, are described, which represent increasingly accurate distribution approximations.

2 Problem Description

Let us assume that the problem is to learn general descriptions of m classes (or concepts) C_1, C_2, …C_m. We assume that training examples of each class are described by *aggregated examples*, which for each attribute specify the mean value, μ , and standard deviation, σ, of this attribute in a subset of instances of this concept.

Let us assume that attributes characterizing the instances are x_1, x_2, .., x_n, and a training set for a class consists of statements in the form of a conjunction:

$$[x_1 = (\mu_1; \sigma_1)] \;\&\; [x_2 = (\mu_2, \sigma_2)] \;\&\; ... \;\&\; [x_n = (\mu_n, \sigma_n)] \qquad (1)$$

in which each condition $[x_i = (\mu_i; \sigma_i)]$ states that in a given subset of training examples the attribute x_i takes the mean value μ_i with the standard deviation σ_i. We will refer to such statements as *aggregated examples*.

A training set for a given class may consists of several such aggregate examples, each characterizing a subset (sample) of specific examples of this class.

3 The Distribution Approximation Approach

Standard methods of machine learning require a set of training examples to learn from. It will not accept statements in the form (1) as examples. The simple idea presented in this paper is to replace each aggregated example by a set of specific examples approximating the distributions of values of each attribute on the basis of the mean and standard deviation for each attribute. Because it is assumed that data do not include a covariance matrix, this information is unknown and the methods assume the independence of the attributes in examples characterizing individual classes.

The following sections present four methods of replacing statements (1) by sets of examples providing consecutively more accurate approximations of a normal distribution that is assumed to have generated the available data.

3.1 Method M0: Mean Only

This is the simplest method for creating a training set based on the statement (1). It reduces (1) to the statement (2), and creates class examples in the form (3).

$$[x_1 = \mu_1] \,\&\, [x_2 = \mu_2] \,\&\, .. \,\&\, .[x_n = \mu_n] \quad (2)$$

$$(\mu_1, \mu_2, ..., \mu_n) \quad (3)$$

In this case, the aggregated training set consists of single vectors representing each subgroup in the given class – one example for each subgroup. The method is very simple. Because it ignores the standard deviation of examples in the subgroup, it does not explore all the available information about the class. This method has been previously applied to analyze metabolic syndrome related data (Wojtusiak et al., 2007).

3.2 Method M1: Mean with Borders

In this method, a training set consists of examples approximating a normal distribution by its mean and selected data points. To explain this matter, let us consider a distribution of values of an attribute, x_i, in the aggregated example (1). Assuming that these values follow a normal distribution $N(\mu, \sigma)$ approximately:

- 68% of values of x_i are within the range $\mu - \sigma \leq x \leq \mu + \sigma$
- 95% of values of x_i, that is 27% more, are within the range $\mu - 2\sigma \leq x \leq \mu + 2\sigma$
- 99.7% of values of x_i, that is 4.7% more are within the range $\mu - 3\sigma \leq x \leq \mu + 3\sigma$

If the size of the population is assumed to be 40, a simple approximation of the normal distribution would be to draw:

- 14 examples with value of x_i between μ and $\mu + \sigma$
- 14 examples with value of x_i between μ and $\mu - \sigma$
- 5 examples with value of x_i between $\mu + \sigma$ and $\mu + 2\sigma$
- 5 examples with value of x_i between $\mu - \sigma$ and $\mu - 2\sigma$
- 1 example with value of x_i between $\mu + 2\sigma$ and $\mu + 3\sigma$
- 1 example with value of x_i between $\mu - 2\sigma$ and $\mu - 3\sigma$

Figure 1 illustrates the difference between this distribution and the normal distribution. It shows the percentages of cases in different ranges according to the normal distribution (2nd row) and corresponding numbers of examples to be drawn within these ranges (3rd row), and their percentages (4th row) in order to approximate the distribution using a population of 40 examples.

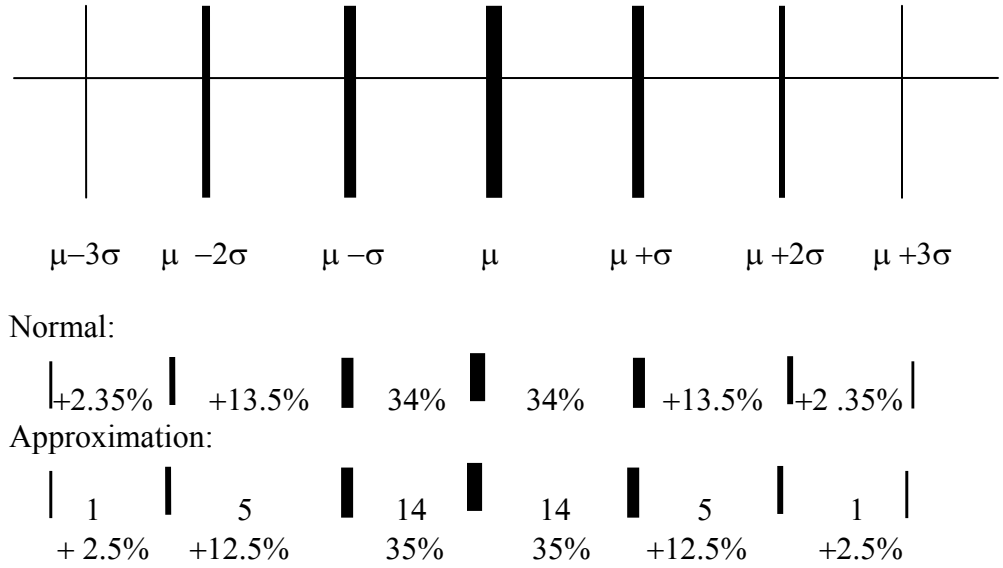


Figure 1: An approximation of the normal distribution by examples drawn from a population of size 40.

Method M1 assumes that the distribution is approximated by seven weighted examples:

- Example 1 with weight 16 and value of $x_i = \mu$
- Example 2 with weight 6 and values of $x_i = \mu + \sigma$
- Example 3 with weight 6 and values of $x_i = \mu - \sigma$
- Example 4 with weight 5 and values of $x_i = \mu + 2\sigma$
- Example 5 with weight 5 and values of $x_i = \mu - 2\sigma$
- Example 6 with weight 1 and values of $x_i = \mu + 3\sigma$
- Example 7 with weight 1 and values of $x_i = \mu - 3\sigma$

This method uses a better approximation of the distribution of examples than Method M0, but that approximation is still very rough. In order to create examples with multiple attributes $x_1, .., x_n$, the process is applied to all attributes producing seven weighted examples:

- $e_1 = (x_1 = \mu_1, ..., x_n = \mu_v)$, weight = 16
- $e_2 = (x_1 = \mu_1 + \sigma_1, ..., x_n = \mu_v + \sigma_n)$, weight = 6
- $e_3 = (x_1 = \mu_1 - \sigma_1, ..., x_n = \mu_v - \sigma_n)$, weight = 6
- $e_4 = (x_1 = \mu_1 + 2\sigma_1, ..., x_n = \mu_v + 2\sigma_n)$, weight = 5
- $e_5 = (x_1 = \mu_1 - 2\sigma_1, ..., x_n = \mu_v - 2\sigma_n)$, weight = 5
- $e_6 = (x_1 = \mu_1 + 3\sigma_1, ..., x_n = \mu_v + 3\sigma_n)$, weight = 1
- $e_7 = (x_1 = \mu_1 - 3\sigma_1, ..., x_n = \mu_v - 3\sigma_n)$, weight = 1

3.3 Method M2: Approximation by Sampling in Ranges (ASR)

In this method, each aggregated example is replaced by a population of 40 examples drawn according to the rules:

- 14 examples are drawn with a randomly generated values $\mu_i \leq x_i < \mu i + \sigma_i$, i=1..n
- 14 examples are drawn with a randomly generated values $\mu_i - \sigma_i < x_i \leq \mu_i$, i=1..n
- 5 examples are drawn with a randomly generated values $\mu_i + \sigma_i \leq x_i < \mu i + 2\sigma_i$, i=1..n
- 5 examples are drawn with a randomly generated values $\mu_i - 2\sigma_i < x_i \leq \mu i + \sigma_i$, i=1..n
- 1 example are drawn with a randomly generated values $\mu_i + 2\sigma_i \leq x_i < \mu i + 3\sigma_i$, i=1..n
- 1 example are drawn with a randomly generated values $\mu_i - 3\sigma_i < x_i \leq \mu i + 2\sigma_i$, i=1..n

As opposed to the method M1, all examples have the same weight of 1. Method M2 is also very simple, as it generates only 40 examples per class, but provides a better approximation than the methods M0 and M1.

3.4 Method M3: Approximation by Sampling Distribution (ASD)

This method replaces an aggregated example by a number of samples, k >> 40 in which values of each variable x_i, i=1,2,..n, are drawn according to the normal distribution $N(\mu_i, \sigma_i)$. To apply this method one could use R statistical package (Venables and Smith, 2008) in which command:

$$x = \text{rnorm}(n, m, s) \tag{4}$$

where n is number of values to be generated, m is the mean, and s is the standard deviation, generates values following normal distribution N(m, s).

This method requires a generation of a large set of training examples (about 1000 for each attribute) than previous methods, but provides the best approximation of the distribution. Thus, this method trades the size of the training set for a better approximation of the distribution.

Because values for each attribute are generated independently, this method produces again only an approximation of the distribution. A better approximation would be created if additional information, such as a covariance matrix, was available.

4 Experimental Comparison

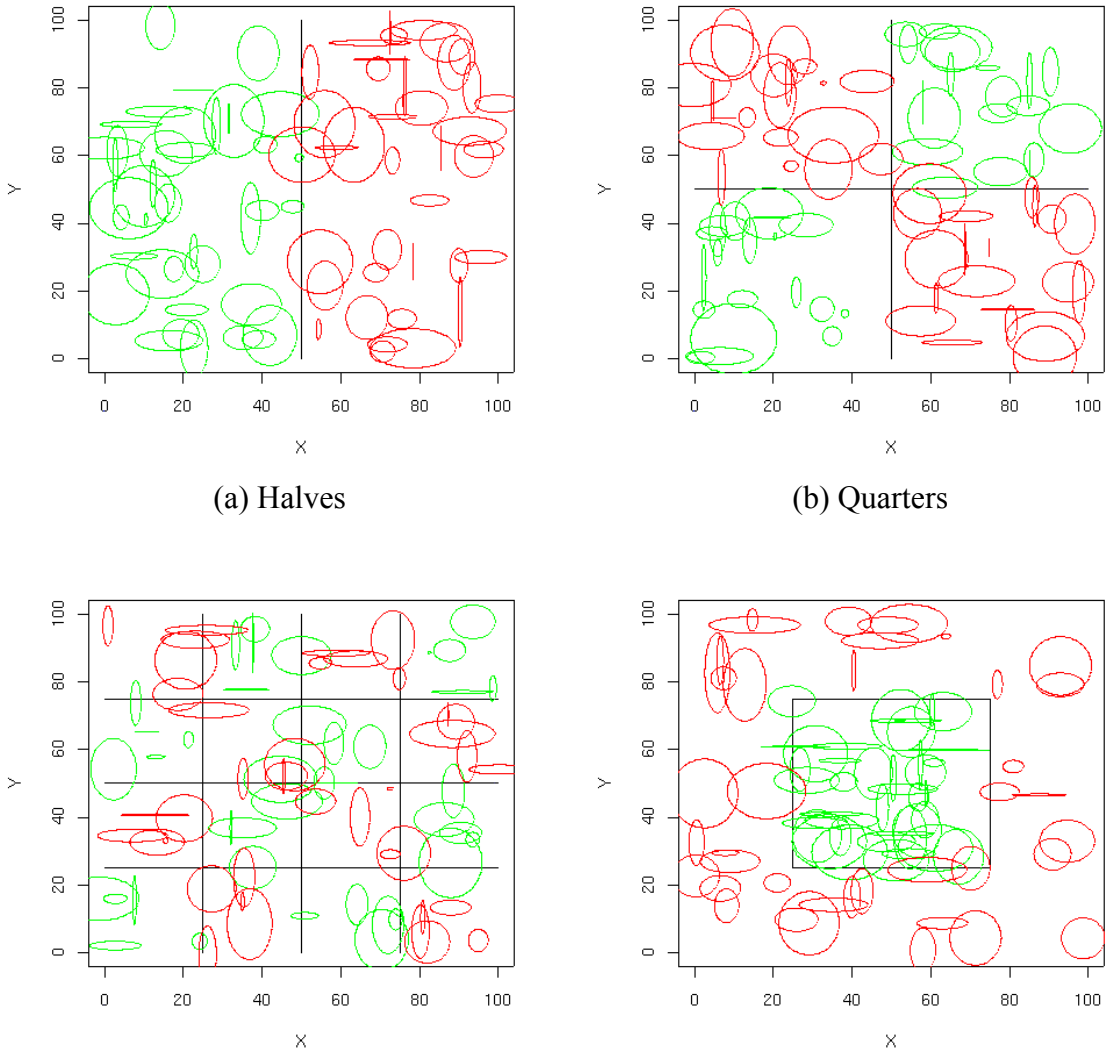
In order to experimentally test and compare the proposed methods, we created a set of four artificial 2-dimensional problems. These problems are simple enough to be graphically illustrated, and complex enough to be used to compare the proposed distribution approximation methods for learning from aggregated data. The following testing procedure was used:

1. Prepare original dataset:
    a. Generate p points $(\mu_{i,x}, \mu_{i,y})$, i=1..p, uniformly distributed in the area representing the concept being learned.
    b. Generate n points $(\mu_{i,x}, \mu_{i,y})$, i=p+1.. p+n+1, uniformly distributed outside the area representing the concept being learned.
    c. For each generated point generate two numbers $\sigma_{i,x} \in [0, \sigma_{xmax}]$ and $\sigma_{i,y} \in [0, \sigma_{ymax}]$ where σ_{xmax} and σ_{ymax} are user-defined parameters.
    d. Generate data points $T_i = (x_{i,j}, y_{i,j})$, $j = 1..n_i$. $x_{i,j}$ are generated according to the normal distribution $N(\mu_{i,x}, \sigma_{i,x})$, and $y_{i,j}$ are generated according to normal distribution $N(\mu_{i,y}, \sigma_{i,y})$, i=1..p+n+1.
    e. Create dataset T by inserting points $(x_{i,j}, y_{i,j}$, 'positive'), i=1..p, and $(x_{i,j}, y_{i,j}$, 'negative'), i=p+1..p+n+1, $j=1..n_i$.
2. Calculate means, $\mu'_{i,x}$, $\mu'_{i,y}$, and standard deviations, $\sigma'_{i,x}$, $\sigma'_{i,y}$ for datasets T_i, i=1..p+n+1.
3. Apply methods M0, M1, M2, and M3 to learn classifiers for the concept on aggregated data in the form $[x_i=(\mu_{i,x}; \sigma_{i,x})]$ & $[y_i=(\mu_{i,x}, \sigma_{i,x})]$, i=1..p+n+1.
4. Test the learned classifiers on the original dataset T.
5. Repeat steps 1 – 4 ten times with different random numbers initialization, and report average results.

In the presented experiments we used the following values: p=20, n=20, and n_i=40, i=1..40. For each example problem, the process was repeated three times with $\sigma_{xmax}=\sigma_{ymax}$=5, 10, 20, respectively.

Four example problems on which the method was experimentally tested are, "halves," "quarters," "checker," and "square." These problems represent different levels of difficulty of learning from aggregated data, because of overlapping data belonging and not belonging to the learned concepts.

The problems are illustrated on Figure 2. In the figure, each ellipse represents mean values and standard deviations of one original dataset. Centers of the ellipses are $(\mu_{i,x}, \mu_{i,y})$, i=1..p, and radii of the ellipses are $\sigma_{i,x}$ and $\sigma_{i,y}$, respectively.



(a) Halves (b) Quarters

(c) Checker (d) Square

Figure 2: Illustration of four problems used to test methods for learning from aggregated data. Each ellipse represents one original distribution with center ($\mu_{i,x}$, $\mu_{i,y}$) and radii $\sigma_{i,x}$ and $\sigma_{i,x}$. Green ellipses represent positive concept examples, and red ellipses represent negative concept examples. The illustrated data were generated with $\sigma_{xmax}=\sigma_{ymax}= 10$.

In the experimental evaluation we compared methods M0, M1, M2, and M3, with results obtained on the original not aggregated dataset. To learn classifiers we applied the AQ21 rule learning program (Wojtusiak, 2004; Wojtusiak et al., 2006). In order to reduce possible bias caused by AQ21's parameters, the program was applied with four different settings of parameters. Two program's modes were compared, namely *theory formation* (TF), in which the program learns complete and consistent classifiers, and *approximate theory formation* (ATF), in which program allows partial inconsistency of learned rules, but optimizes the quality measure *Q(w)*. The measure represents a tradeoff between

rule's confidence gain and completeness, as indicated by a parameter *w*. For details of the measure please refer, for example, to the description by Kaufman and Michalski (2000). Three values of the parameter w were used: 0.1, 0.3, and 0.5.

Table 1 compares results of testing classifiers learned by AQ21 on datasets created using methods M0, M1, M2, and M3, with the original data. The results are averaged over different AQ21 settings. Not averaged values for the case of the "quarters" problem with standard deviation set to 5 are shown in Table 2. All results are reported in terms of accuracies and precisions (Wojtusiak, 2004) of learned classifiers as tested on the original data. The results presented in Table 1 clearly indicate that application of methods M0, M1, M2, and M3 gives consecutively better accuracies. These accuracies are, however, in most cases worse than those obtained by classifiers learned from the original data.

The measure of precision is used to capture the situation when the program is unable do give a definitive answer to which class a testing example should belong. In many situations it is better to give an imprecise answer, than a precise incorrect classification. Details of the measure are described by Wojtusiak (2004).

Table 1: Accuracies and precisions obtained by methods M0, M1, M2, and M3, from aggregated data and on the original data for four tested problems. The original data was generated with three values of standard deviation: 5, 10, and 20.

	Halves		**Quarters**		**Checker**		**Square**		**Average**	
Standard deviation = 5										
	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	**Acc.**	**Prec.**
Original	1.00	0.83	0.98	0.76	0.98	0.76	0.99	0.70	**0.99**	**0.76**
M0	0.98	1.00	0.95	0.98	0.95	0.98	0.96	0.95	**0.96**	**0.97**
M1	0.99	0.80	0.96	0.74	0.96	0.74	0.97	0.68	**0.97**	**0.74**
M2	0.99	0.81	0.97	0.70	0.97	0.70	0.99	0.68	**0.98**	**0.72**
M3	1.00	0.82	0.97	0.72	0.97	0.72	0.99	0.67	**0.98**	**0.73**
Standard deviation = 10										
Original	0.99	0.73	0.98	0.60	0.98	0.60	0.98	0.62	**0.98**	**0.64**
M0	0.96	1.00	0.90	0.97	0.90	0.97	0.92	0.94	**0.92**	**0.97**
M1	0.98	0.67	0.91	0.66	0.91	0.66	0.94	0.56	**0.93**	**0.64**
M2	0.98	0.71	0.94	0.63	0.94	0.63	0.95	0.63	**0.95**	**0.65**
M3	0.99	0.73	0.96	0.60	0.96	0.60	0.96	0.61	**0.96**	**0.63**
Standard deviation = 20										
Original	0.98	0.65	0.94	0.56	0.94	0.56	0.95	0.53	**0.95**	**0.58**
M0	0.91	0.99	0.82	0.93	0.82	0.93	0.83	0.91	**0.85**	**0.94**
M1	0.93	0.59	0.84	0.56	0.84	0.56	0.86	0.49	**0.87**	**0.55**
M2	0.95	0.56	0.89	0.56	0.89	0.56	0.88	0.55	**0.90**	**0.56**
M3	0.96	0.59	0.89	0.52	0.89	0.52	0.89	0.53	**0.91**	**0.54**

Table 2: Accuracies and precisions obtained in single AQ21 executions with different methods of handling aggregated values on the "quarters" problem with standard deviation equal 5. With each combination of parameters program was executed 10 times as indicated by the numbers 0..9. Program parameters indicate mode of execution (ATF or TF) and the number accompanying ATF mode is a weight, w, of confidence gain vs. completeness of rules. "Org." indicates original, not aggregated, dataset.

			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Avg.**	**SD**
ATF 01	Org.	Acc.	0.98	0.94	0.99	0.98	0.96	0.94	0.95	0.96	0.98	0.87	0.95	0.03
		Prec.	0.99	0.98	0.89	0.71	0.99	0.94	0.84	0.98	0.83	1.00	0.91	0.10
	M0	Acc.	0.96	0.92	0.93	0.93	0.96	0.93	0.94	0.96	0.95	0.92	0.94	0.02
		Prec.	0.97	0.98	0.99	0.99	0.99	0.99	0.99	1.00	1.00	0.97	0.99	0.01
	M1	Acc.	0.98	0.90	0.93	0.95	0.93	0.93	0.94	0.90	0.90	0.86	0.92	0.03
		Prec.	0.96	0.98	0.93	0.96	0.96	0.83	0.93	0.99	0.92	0.90	0.94	0.05
	M2	Acc.	0.97	0.95	0.98	0.98	0.97	0.97	0.96	0.93	0.95	0.89	0.95	0.03
		Prec.	0.98	0.85	0.80	0.58	0.76	0.65	0.89	1.00	0.89	0.96	0.83	0.14
	M3	Acc.	0.98	0.97	0.97	0.96	0.96	0.95	0.95	0.93	0.98	0.80	0.94	0.05
		Prec.	0.96	0.93	0.90	1.00	0.99	0.78	0.92	0.96	0.82	0.97	0.92	0.07
ATF 03	Org.	Acc.	1.00	0.99	0.99	0.99	0.99	0.99	0.97	0.97	0.99	0.96	0.99	0.01
		Prec.	0.68	0.90	0.66	0.62	0.72	0.37	0.76	0.92	0.73	0.49	0.68	0.17
	M0	Acc.	0.97	0.93	0.95	0.95	0.96	0.93	0.94	0.96	0.95	0.93	0.95	0.02
		Prec.	0.99	0.98	0.96	0.94	0.99	0.97	0.99	1.00	1.00	1.00	0.98	0.02
	M1	Acc.	0.97	0.99	0.97	0.99	0.97	0.96	0.97	0.98	0.99	0.94	0.97	0.02
		Prec.	0.96	0.43	0.69	0.44	0.72	0.69	0.54	0.61	0.64	0.70	0.64	0.15
	M2	Acc.	0.99	0.97	0.98	1.00	0.99	0.99	0.98	0.96	0.98	0.95	0.98	0.01
		Prec.	0.57	0.70	0.60	0.47	0.60	0.57	0.58	0.81	0.49	0.39	0.58	0.12
	M3	Acc.	0.99	0.98	0.99	0.99	1.00	0.99	0.97	0.98	0.97	0.99	0.98	0.01
		Prec.	0.91	0.74	0.47	0.46	0.49	0.46	0.74	0.73	0.71	0.38	0.61	0.18
ATF 05	Org.	Acc.	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.00
		Prec.	0.76	0.57	0.53	0.27	0.41	0.35	0.26	0.51	0.47	0.39	0.45	0.15
	M0	Acc.	0.97	0.94	0.95	0.97	0.96	0.93	0.94	0.96	0.96	0.92	0.95	0.02
		Prec.	1.00	0.96	0.95	0.63	0.99	0.97	0.99	1.00	0.95	1.00	0.94	0.11
	M1	Acc.	0.99	0.97	1.00	1.00	0.99	0.97	0.98	0.99	1.00	0.98	0.99	0.01
		Prec.	0.40	0.62	0.27	0.22	0.58	0.62	0.42	0.30	0.57	0.34	0.43	0.15
	M2	Acc.	1.00	0.99	1.00	1.00	0.98	0.99	0.98	0.99	1.00	0.97	0.99	0.01
		Prec.	0.68	0.34	0.33	0.26	0.60	0.30	0.38	0.58	0.26	0.46	0.42	0.15
	M3	Acc.	0.99	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.00
		Prec.	0.59	0.46	0.38	0.22	0.30	0.29	0.16	0.56	0.57	0.23	0.38	0.16
TF	Org.	Acc.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
		Prec.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
	M0	Acc.	0.99	0.96	0.96	0.97	0.96	0.94	0.95	0.96	0.96	0.93	0.96	0.02
		Prec.	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.01
	M1	Acc.	0.98	0.95	0.96	0.95	0.95	0.94	0.94	0.94	0.93	0.88	0.94	0.02
		Prec.	0.98	0.94	0.95	0.93	0.94	0.92	0.94	0.94	0.94	0.92	0.94	0.02
	M2	Acc.	0.99	0.98	0.98	0.97	0.97	0.96	0.96	0.97	0.97	0.93	0.97	0.01
		Prec.	0.99	0.97	0.98	0.97	0.97	0.95	0.98	0.98	0.96	0.96	0.97	0.01
	M3	Acc.	0.99	0.98	0.98	0.98	0.98	0.97	0.97	0.98	0.98	0.95	0.98	0.01
		Prec.	0.99	0.99	0.98	0.98	0.98	0.98	0.98	0.99	0.98	0.96	0.98	0.01

5 Conclusion

The described methods, M0, M1, M2, and M3 generate consecutively better approximations of the normal distributions of attribute values. They are used to create subsets of training datasets for a learning program when the original data are available only in an aggregated form. These approximations require, however, a consecutively larger number of examples to generate. The methods are simple and allow one to apply any existing learning program without modification. Experimental evaluation using AQ21 program on four example problems confirms that better approximations lead to better predictive accuracies.

References

Kaufman, K. and Michalski, R. S., "An Adjustable Rule Learner for Pattern Discovery Using the AQ Methodology," Journal of Intelligent Information Systems, 14, pp 199-216, 2000.

Michalski, R. S., "ATTRIBUTIONAL CALCULUS: A Logic and Representation Language for Natural Induction," *Reports of the Machine Learning and Inference Laboratory*, MLI 04-2, George Mason University, Fairfax, VA, April, 2004.

Wojtusiak, J., "AQ21 User's Guide," *Reports of the Machine Learning and Inference Laboratory*, MLI 04-3, George Mason University, Fairfax, VA, September, 2004 (updated in September, 2005).

Wojtusiak, J., Michalski, R. S., Kaufman, K. and Pietrzykowski, J., "The AQ21 Natural Induction Program for Pattern Discovery: Initial Version and its Novel Features," *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, Washington D.C., November 13-15, 2006.

Venables, W. N., Smith D. M., and the R Development Core Team, "An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics Version 2.7.0 (2008-04-22)," Available from http://www.r-project.org/, 2008.

A publication of the *Machine Learning and Inference Laboratory*
George Mason University
Fairfax, VA 22030-4444 U.S.A.
http://www.mli.gmu.edu

Editor: Janusz Wojtusiak

The *Machine Learning and Inference (MLI) Laboratory Reports* are an official publication of the Machine Learning and Inference Laboratory, which has been published continuously since 1971 by R.S. Michalski's research group (until 1987, while the group was at the University of Illinois, they were called ISG (Intelligent Systems Group) Reports, or were part of the Department of Computer Science Reports).