An intelligent learning machine

S. Sayad¹,², S. T. Balke¹,² & S. Sayad²
¹University of Toronto, Canada
²iSmartsoft Inc., Canada

Abstract

In this paper we present a new learning system, the “Intelligent Learning Machine” (ILM). We associate intelligence with the power to learn, forget, grow, contract, interact, and co-operate incrementally, on-line, and in real time. Intelligence in the ILM is based upon the use of a specially customized weight table. The ILM enables parallel data processing and it is well suited to a wide variety of applications and promises unprecedented performance gains in dynamic environments. Here we show how Linear and Non-linear Regression and Classification modeling methods are transformed into intelligent methods. This method has now been successfully software implemented and tested using a variety of databases. Hardware implementation of the ILM is feasible and we foresee an ILM chip for faster computations and mobile applications. Subsequent papers will show how the ILM can be applied to methods such as Bayesian Models, Markov Chain, Hidden Markov Models, Linear Discriminant Analysis, Association Rules, OneR, Principal Component Analysis and Linear Support Vector Machines.

1 Introduction

The objective of this paper is to present a new learning system which we term the “Intelligent Learning Machine” (ILM) [8]. Learning in nature is incremental, on-line and in real time. However, most learning algorithms in AI and Machine Learning operate in a batch mode where having all of the relevant data at once is a requirement. Furthermore, those that are not batch, such as incremental learning methods [1,3,4,6], on-line learning methods [2,9,11], and real-time learning methods [7,10] are all very constrained in the degree of intelligence they can exhibit. Machine learning needs a system that can learn, forget, grow, contract, interact, and co-operate incrementally, online and in real time. Speed,
portability and ease of implementation are important. It must easily be implemented on any system, from an intelligent agent to a micro-robot. The ILM described here satisfies all of these requirements and it does so by using a specially customized weight table. “Automatic summary tables” are well known in database technology. Efficiently updating summary tables in an incremental way as new data becomes available for the database has been the subject of many publications [5,12]. However, these summary tables are buried in huge databases and are manipulated with the objective of responding to a wide variety of possible queries. They cannot meet the requirements of intelligent learning mentioned above.

The ILM converts many ordinary learning methods into intelligent learning methods. In contrast to other learning methods which have two phases, learning and prediction, the ILM involves three phases: learning, modeling, and prediction. The ILM is linearly scalable and is a rigorous method for making parallel data processing readily accomplished, especially for very large databases. Any size dataset can be divided to smaller parts and each part can be processed separately. The results can then be joined together to obtain the same model as if we had processed the whole dataset at once.

Figure 1: The Intelligent Learning Machine (ILM).

2 The Intelligent Learning Machine

Figure 1 shows a diagram of the Intelligent Learning Machine (ILM). Data is collected as it is generated, passed to a learner that formulates and updates a table, the ILM Weight Table (IWT). This single table fulfills the needs of many different modeling methods. In regression analysis, for example, the Modeller, uses the entries in this table to obtain equation coefficients. In Bayesian Classification it computes discriminant functions. [Surprisingly, this one table is also common to Non-linear Regression, Linear and Non-linear Classification, Bayesian Models, Markov Chain, Hidden Markov Models, Linear Discriminant Analysis, Association Rules, OneR, Principal Component Analysis and Linear Support Vector Machines] A user interface can then request a prediction from the model. The properties of the IWT impart powerful incremental learning capabilities along with a parallel data processing capability to whatever modeling method is used with it. These methods can then learn, forget, grow, contract, interact, and co-operate incrementally, online and in real time. It is also fast and easy to implement. The central part of the ILM, the IWT and its properties are described in the following sections.
2.1 The ILM weight table

The ILM Weight Table, or IWT, is a two dimensional table (Table 1). In this table, n is the number of input (independent) variables and m is the number of output (dependent) variables. The definition of input and output variables is dynamic in the IWT. That is, a variable can be selected as an input in one situation and an output in another situation. The IWT can have a mixture of numerical or categorical variables. It is the flexibility provided by the IWT, which is responsible for the intelligence in the resulting models.

Table 1: The general structure of IWT.

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>...</th>
<th>$X_j$</th>
<th>...</th>
<th>$X_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$W_{11}$</td>
<td>...</td>
<td>$W_{1j}$</td>
<td>...</td>
<td>$W_{1n}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_i$</td>
<td>$W_{il}$</td>
<td>...</td>
<td>$W_{ij}$</td>
<td>...</td>
<td>$W_{in}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_m$</td>
<td>$W_{ml}$</td>
<td>...</td>
<td>$W_{mj}$</td>
<td>...</td>
<td>$W_{mn}$</td>
</tr>
</tbody>
</table>

As shown in Table 2, the basic unit of the IWT includes an input variable $X_j$, an output variable $X_i$ and a weight $W_{ij}$:

Table 2: The basic unit of IWT.

<table>
<thead>
<tr>
<th></th>
<th>$X_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_j$</td>
<td>$W_{ij}$</td>
</tr>
</tbody>
</table>

where $W_{ij}$ includes one or more of the following four basic elements:

- $N_{ij}$ is the total number of data values
- $\Sigma X_i$ is the sum of variable $X_i$
- $\Sigma X_j$ is the sum of variable $X_j$
- $\Sigma X_iX_j$ is the sum of multiplication of variable $X_i$ and $X_j$

All four elements can be updated incrementally, online and in real time. Also, there are more elements, which could be included in the weight table such
as $\Sigma X^2$, $\Sigma X^3$, and $\Sigma X^4$. One important feature of the IWT is that the basic units ($W_{ij}$) are independent and can have different sources of data.

### 2.2 Learning in the ILM

Learning in the ILM is incremental, online, and in real time. It can be done by adding the new weight elements $W_{ij}^{\text{record}}$ to the existing one $W_{ij}^{\text{old}}$:

$$W_{ij}^{\text{new}} = W_{ij}^{\text{old}} + W_{ij}^{\text{record}}$$

(1)

Suppose we have a model with two variables ($X_1, X_2$) and we want to learn a new record of data $R=\{x_1, x_2\}$ where $x_1$ and $x_2$ are the values of $X_1$ and $X_2$. There are just two simple steps:

**Step 1:** Create a weight table for the new record.

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$x_1$</td>
<td>$x_1$</td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>$x_2$</td>
</tr>
<tr>
<td></td>
<td>$x_1x_1$</td>
<td>$x_1x_1$</td>
</tr>
</tbody>
</table>

**Step 2:** Add the weight table from step 1 to the existing weight table.

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$N_{ij}+1$</td>
<td>$N_{ij}+1$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_1 + x_1$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_1 + x_1$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_1 X_1 + x_1 x_1$</td>
<td>$\Sigma X_1 X_1 + x_1 x_1$</td>
</tr>
<tr>
<td>$X_2$</td>
<td>$N_{2j}+1$</td>
<td>$N_{2j}+1$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 + x_2$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_1 + x_1$</td>
</tr>
<tr>
<td></td>
<td>$\Sigma X_2 X_1 + x_2 x_1$</td>
<td>$\Sigma X_2 X_1 + x_2 x_1$</td>
</tr>
</tbody>
</table>

If desired, a set of records can be added at one time instead of as a single record.
2.3 Forgetting in the ILM

Forgetting is as simple as learning but instead of adding we subtract the new weight elements $W_{ij}^{\text{record}}$ from the existing one $W_{ij}^{\text{old}}$.

$$W_{ij}^{\text{new}} = W_{ij}^{\text{old}} - W_{ij}^{\text{record}}$$

If desired, a set of records can be subtracted/added at one time instead of a single record.

2.4 Growing in the ILM

It is important to have a learning structure, which can grow incrementally, online, and in real-time. We can readily add new variables to the IWT without any adverse consequences.

$$W_m \rightarrow W_{(m+1)\times(n+1)}$$

The Eqn. (3) shows that the ILM grows by adding a new column and a new row for the new variable to the existing IWT with the weight elements equal to zero. Suppose we have a model with just two variables $(X_1, X_2)$ and want to add a new record of data that has values for three variables $R=\{x_1, x_2, x_3\}$. There are three steps used to accomplish this task:

**Step 1:** Create a weight table for the new record.

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x_1</td>
<td>x_1</td>
<td>x_1</td>
<td>x_1</td>
</tr>
<tr>
<td>x_1</td>
<td>x_2</td>
<td>x_3</td>
<td>x_3</td>
</tr>
<tr>
<td>x_1 x_1</td>
<td>x_1 x_2</td>
<td>x_1 x_3</td>
<td></td>
</tr>
<tr>
<td>$X_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x_2</td>
<td>x_2</td>
<td>x_2</td>
<td>x_2</td>
</tr>
<tr>
<td>x_1</td>
<td>x_2</td>
<td>x_3</td>
<td>x_3</td>
</tr>
<tr>
<td>x_2 x_1</td>
<td>x_2 x_2</td>
<td>x_2 x_3</td>
<td></td>
</tr>
<tr>
<td>$X_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x_3</td>
<td>x_3</td>
<td>x_3</td>
<td>x_3</td>
</tr>
<tr>
<td>x_1</td>
<td>x_2</td>
<td>x_3</td>
<td>x_3</td>
</tr>
<tr>
<td>x_3 x_1</td>
<td>x_3 x_2</td>
<td>x_3 x_3</td>
<td></td>
</tr>
</tbody>
</table>
Step 2: Add a new column and a new row for the new variable to the existing IWT with the weight elements equal to zero.

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{11}$</td>
<td>$N_{12}$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_1$</td>
<td>$\Sigma X_1$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_1$</td>
<td>$\Sigma X_2$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_1 X_1$</td>
<td>$\Sigma X_1 X_2$</td>
<td>0</td>
</tr>
<tr>
<td>$N_{21}$</td>
<td>$N_{22}$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_2$</td>
<td>$\Sigma X_2$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_2$</td>
<td>$\Sigma X_2$</td>
<td>0</td>
</tr>
<tr>
<td>$\Sigma X_2 X_1$</td>
<td>$\Sigma X_2 X_2$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Step 3: Update the existing weight table.

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{11} + 1$</td>
<td>$N_{12} + 1$</td>
<td>$N_{13} = 0 + 1$</td>
</tr>
<tr>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_1 = 0 + x_1$</td>
</tr>
<tr>
<td>$\Sigma X_1 + x_1$</td>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 = 0 + x_3$</td>
</tr>
<tr>
<td>$\Sigma X_1 X_1 + x_1 x_1$</td>
<td>$\Sigma X_1 X_2 + x_1 x_2$</td>
<td>$\Sigma X_1 X_2 = 0 + x_1 x_3$</td>
</tr>
<tr>
<td>$N_{21} + 1$</td>
<td>$N_{22} + 1$</td>
<td>$N_{22} = 0 + 1$</td>
</tr>
<tr>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 = 0 + x_2$</td>
</tr>
<tr>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 + x_2$</td>
<td>$\Sigma X_2 = 0 + x_3$</td>
</tr>
<tr>
<td>$\Sigma X_2 X_1 + x_2 x_1$</td>
<td>$\Sigma X_2 X_2 + x_2 x_2$</td>
<td>$\Sigma X_2 X_2 = 0 + x_2 x_3$</td>
</tr>
<tr>
<td>$N_{31} = 0 + 1$</td>
<td>$N_{32} = 0 + 1$</td>
<td>$N_{31} = 0 + 1$</td>
</tr>
<tr>
<td>$\Sigma X_3 = 0 + x_3$</td>
<td>$\Sigma X_3 = 0 + x_3$</td>
<td>$\Sigma X_3 = 0 + x_3$</td>
</tr>
<tr>
<td>$\Sigma X_3 = 0 + x_3$</td>
<td>$\Sigma X_3 = 0 + x_3$</td>
<td>$\Sigma X_3 = 0 + x_3$</td>
</tr>
<tr>
<td>$\Sigma X_3 X_1 + x_3 x_1$</td>
<td>$\Sigma X_3 X_2 + x_3 x_2$</td>
<td>$\Sigma X_3 X_2 = 0 + x_3 x_3$</td>
</tr>
</tbody>
</table>

2.5 Contracting in the ILM

Not only having a learning structure that can grow intelligently is important but also we want to be able to discard useless variables without difficulty.

$$W^{\text{max}} \rightarrow W^{(m-1)<(n-1)}$$

(4)
Eqn. (4) shows that the ILM is contracted by removing the column and the row related to the discarded variable.

2.6 Interacting in the ILM: dynamic modeling

An intelligent learning model should be adaptive: it should dynamically adjust its strategy to take into account the behavior of particular problem to be solved and to change its internal structure based on different set of inputs.

\[
W^{pxq} \subseteq W^{mxn} \quad p \leq m, \quad q \leq n
\]  

Using the ILM dynamic modeling ability, hundreds of models can be generated and tested in a very short period of time.

Suppose we have a model with three variables \((X_1, X_2, X_3)\) but there are just two variables \((X_1, X_3)\) having values and the value for the \(X_2\) is missing (e.g., a sensor becomes disconnected in a robot). The model for three variables is not valid for just two variables but because of the flexible structure of the ILM it can be easily sliced out to make a temporary weight table for just two variables. A new model can be generated just for two variables \((X_1, X_3)\) and all of this can be done on-line and in real-time.

2.7 Co-operation in the ILM

Another feature of the ILM is that it can enable different learning models to join (temporarily or permanently) to form a super model. This is a crucial feature for distributed and multi-agents systems.

\[
W^{pxq} + W^{pxq} \rightarrow W^{mxn} \quad p, s \leq m, \quad q, r \leq n
\]

Suppose we have two intelligent learning machines, A and B with two variables \((X_1, X_2)\). The joined weight table is the sum of two weight tables (Table 3).

Table 3: A joined weight table in ILM.

<table>
<thead>
<tr>
<th></th>
<th>(X_1)</th>
<th>(X_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>(W_{11}^A + W_{11}^B)</td>
<td>(W_{12}^A + W_{12}^B)</td>
</tr>
<tr>
<td>(X_2)</td>
<td>(W_{21}^A + W_{21}^B)</td>
<td>(W_{22}^A + W_{22}^B)</td>
</tr>
</tbody>
</table>
3 The ILM modeling phase

One important feature in the ILM is separation of the modeling phase from the learning and predicting phases. This allows many learning algorithms to use the same weight table and inherit the intelligent features. In the following paragraph we show how this is done with two methods, Linear Regression and Linear Classification, to result in what we term “Intelligent Linear Regression” and “Intelligent Linear Classification”.

3.1 Intelligent linear regression

There are six steps to generate an intelligent linear regression equation from the ILM weight table. Coefficients $\beta$ are estimated by the least square estimators $b$ which are also maximum likelihood estimators and among all unbiased estimators they have the smallest standard errors. These steps are as follows:

**Step 1:** Calculate the covariance matrix. (Note: if $i = j$ the covariance is the variance.)

$$
\text{Covar}_{ij} = \frac{\sum X_iX_j - \frac{1}{N} \left( \sum X_i \sum X_j \right)}{N_{ij}}
$$

**Step 2:** Calculate the correlation matrix from the covariance matrix. (Note: if $i = j$ the elements of the matrix are unity.)

$$
R_{ij} = \frac{\text{Covar}_{ij}}{\sqrt{\text{Var}_i \times \text{Var}_j}}
$$

where:

$$
\text{Var}_i = \text{Covar}_{ii}, \quad \text{Var}_j = \text{Covar}_{jj}
$$

**Step 3:** Select one of the variables, $X_j$, as the dependent variable and refer to it as $Y$, then slice the correlation matrix to a matrix for the independent variables $R_{i,j}$.
and a vector for the dependent variable $\mathbf{R}_{y,j}$. Calculate the coefficient $\beta$ for independent variables.

$$\beta_j = \mathbf{R}_{i,j}^T \mathbf{R}_{y,j}$$

**Step 4:** Calculate sample coefficients $b_j$

$$b_j = \beta_j \left( s_y / s_j \right)$$

$s_y$ is the sample estimate of the standard deviation of the dependent variable and $s_j$ the sample estimate of the standard deviation of the independent variables. These quantities are readily calculated from the ILM weight table.

**Step 5:** Calculate intercept $a$

$$a = \overline{Y} - b_1 \overline{X_1} - b_2 \overline{X_2} - \ldots - b_n \overline{X_n}$$

where any mean value can be calculated from $\Sigma X_i / N_i$

**Step 6:** Finally the linear equation, which can be used for the prediction, is:

$$Y = a + b_1 X_1 + b_2 X_2 + \ldots + b_n X_n$$

### 3.2 Intelligent linear classification

We use classification when the output (dependent) variable is a categorical variable. Suppose the dependent variable $X_d$ has $k$ values. Instead of just one regression model we build $k$ models by using the same procedure as for intelligent linear regression.

$$X_{di} = a_i + b_{i1} X_{i1} + b_{i2} X_{i2} + \ldots + b_{in} X_{in}$$

$$X_{dj} = a_j + b_{j1} X_{j1} + b_{j2} X_{j2} + \ldots + b_{jn} X_{jn}$$

$$\ldots$$

$$X_{dk} = a_k + b_{k1} X_{k1} + b_{k2} X_{k2} + \ldots + b_{kn} X_{kn}$$

In the prediction phase, these $k$ models compete with each other and the model with the highest value will be the winner.

### 4 The ILM prediction phase

As we mentioned earlier, one important feature in the ILM is separation of the modeling phase from the learning and predicting phases. With a trained ILM we can use many learning algorithms, which rely on the IWT and therefore inherit the intelligent features. Another important feature of the ILM in the prediction phase is the ability to create hundreds of different models in just a few seconds. It is possible to have different models and for any model having different set of
input and output variables. Computations are performed on the IWT and not on the original data.

4.1 Data compression

A dataset with $10^6$ records and with 10 variables which includes just numbers requires at least $10^6 \times 10 \times 8 = 8 \times 10^7$ bytes or 80 mega bytes memory. The ILM weight table for this dataset is just 3200 bytes ($10 \times 10 \times 4 \times 8$) which gives a compression rate of more than 20,000 times!

4.2 Processing time

If it requires about 30 seconds to construct a traditional regression model for a dataset with $10^6$ records and 10 dimensions using a desktop computer (Intel Pentium III, 1 GHz with 256 MB memory) it takes almost the same period of time to construct an intelligent regression model based on the ILM. However, any changes to the structure of the dataset such as adding or removing new records or variables or selecting another dependent variables will result in spending another 30 seconds for the traditional model. In contrast, an intelligent model needs just a few milliseconds to process the ILM weight table.

5 Conclusions

The Intelligent Learning Machine (ILM) is new, powerful, and yet disarmingly simple in appearance. The method transforms ordinary learning methods (e.g., Linear and Non-linear Regression, Linear and Non-linear Classification, Bayesian Models, Markov Chain, Hidden Markov Models, Linear Discriminant Analysis, Association Rules, OneR rule, Principal Component Analysis and Linear Support Vector Machines) into intelligent methods. That is, it creates models that have the power to learn, forget, grow, contract, interact, and co-operate, incrementally, online, and in real time. This intelligence also includes speed and ease of implementation. Furthermore, the ILM is a rigorous method for making parallel data processing readily accomplished, especially for very large databases. Any size dataset can be divided to smaller parts and each part can be processed separately. The results can then be joined together to obtain the same model as if we had the whole dataset at once. The intelligent models obtained from the ILM are linearly scalable methods. Searching for the best model conventionally requires building and testing many different models, sometimes numbering in the hundreds, before the best solution can be found. Using the ILM dynamic modeling ability, hundreds of models can be generated and tested in a very short period of time. In this paper we have shown how linear/non-linear regression and classification are transformed into intelligent methods. In future papers we will show how the ILM can be applied to provide many other intelligent learning methods. This method has now been successfully software implemented and tested using a variety of databases. Hardware implementation of the ILM is feasible and we foresee an ILM chip in future for mobile applications.


References


