CHAPTER 5

ONLINE DYNAMIC REGRESSIVE LEARNING

The dynamic CEP becomes more efficient when the patterns of interest are learned automatically from the streaming data, and the newly learned patterns are added to the engine. Hence this chapter proposes an incremental Online Dynamic Regressive Learning (ODReL) algorithm. Machine learning is used to learn the properties or characteristics of any system from history as well as continuous observation of the various parameters of the system. Traditional static learning algorithms model the system during the training period with the available data. These algorithms do not provide a scope for adaptability to gradual changes in the parameters characterizing the system that is to be learned. The existing incremental learning systems are computation intensive and do not support self correction using feedback. This chapter proposes a novel incremental learning algorithm, Online Dynamic Regressive Learning (ODReL), which overcomes the disadvantages of the existing system. The proposed ODReL is applied to the prediction of the health parameters and detection of the abnormalities in the vital health parameters of a person.

This chapter presents the proposed ODReL algorithm in detail. Section 5.1 gives an introduction to the incremental learning. Section 5.2 explains the proposed ODReL algorithm and section 5.3 presents the results and performance of ODReL and section 5.4 presents the summary.
5.1 INCREMENTAL LEARNING

Incremental learning is a type of machine learning in which the learning process takes place whenever new example(s) emerge and adjusts what has been learned according to the new example(s). The most prominent difference between incremental learning and traditional machine learning is that it does not assume the availability of a sufficient training set before the learning process, but the training examples appear over time. Incremental learning has recently attracted growing attention from both academia and industry.

There are at least two main reasons why incremental learning is important.

1. From data mining perspective, many of today’s data-intensive computing applications require the learning algorithm to be capable of learning incrementally from large-scale dynamic stream data, and to build up the knowledge base over time to benefit future learning and decision-making process.

2. From the machine intelligence perspective, intelligent systems are able to learn information incrementally. Thus incremental learning becomes an automatic method of preference when all the required data is not available at the same time and when the system is required to initiate the learning process as soon as the first set of data is available.
The main advantages of using an incremental learning algorithm are

- Easy failure handling – due to feedback.
- Entire history is being reused.
- Speculative execution is avoided.
- Designed for low cost commodity hardware.
- Self correcting mechanism.

5.2 ONLINE DYNAMIC REGRESSIVE LEARNING ALGORITHM

The learning procedure is employed to enable the system to adapt itself to the particular patient. The proposed Online Dynamic Regressive Learning algorithm does the functions of abnormality detection and prediction of the next value of the parameter, using the previous instances of the parameter values and initial history. The ODReL algorithm supports correction with the learned knowledge using the update and feedback from the doctor.

Input to the algorithm is health parameter vector \([a_t]\). In the algorithm \(ShortValue\) represents the mean value of the particular parameter over a short window, say 15, and \(LongValue\) represents the mean value of the parameter from the beginning of history. Thus, \(ShortValue\) represents the current trend of the patient while the \(LongValue\) represents the normalized health history of the patient. On receiving the health parameter, first the validity of the parameter is checked then the \(ShortValue\) and \(LongValue\) are computed.
ODReL algorithm is given below.

Data: Health Parameter Vector \([ a_i ]\)

Result: Updated Health Parameter Vector \([ a_{i+1} ]\)

Aggregation:

while \(Aggregation valide\) do

\[
\text{LongValue}_i \leftarrow \sum_{j=0}^{i} a_j / i; \\
\text{ShortValue}_i \leftarrow \sum_{k=i-15}^{i} a_k / 15; \\
\text{DiffThresh} \leftarrow |(\text{LongValue}_i + \text{ShortValue}_i) / 2 - ((a_{\text{max}} + a_{\text{min}}) / 2)|
\]

if \((\text{ShortValue}_i \sim \text{LongValue}_i) \lor \text{DiffThresh} > \text{DiffThresh}\) then

Abnormality;

Backup and Feedback;

else

Update \(a_i\);

end

getSensitivity();

\(\alpha \leftarrow \text{SensitivityRange}/\text{LongValue}\)

\(a_{i+1} \leftarrow a_i \cdot \alpha + \text{historyFactor} \cdot a_i\)

reconfigure \(\alpha, \text{historyFactor}\)

end

Abnormality Detection

For detecting the abnormality following formula is used.

\[
( | \text{ShortValue} - \text{LongValue} | > \text{DiffThresh} ) \quad \text{AND} \quad ( a_i > a_{\text{minthresh}} \lor a_i < a_{\text{maxthresh}} )
\] (5.1)

Here, \(a_i\) is the value of the parameter in the current instance.

\(a_{\text{minthresh}}, a_{\text{maxthresh}}\) are the threshold values of a given parameter.

\(\text{DiffThresh}\) represents the expected difference between any two points in the given normal large sample data set. Hence, when the difference between the \(\text{LongValue}\) (point A) and \(\text{ShortValue}\) (point B) is greater than the \(\text{DiffThresh}\), it intuitively means that the large sample contains data which are
not a part of a normal data. Since, for time t-1, the data belonged to a normal large sample, by the principle of elimination, the last data is proved to be the abnormal data. Hence the $\text{DiffThresh}$ provides the sufficient condition for an abnormality. $\text{DiffThresh}$ is calculated as follows.

$$\text{DiffThresh} = \left| \frac{(\text{ShortValue} + \text{LongValue})}{2} - \frac{(a_{\text{min}} + a_{\text{max}})}{2} \right|$$

(5.2)

Where $\text{LongValue}$ is the mean of the large sample data set and $\text{ShortValue}$ is the mean of the short window and $a_{\text{max}}$ and $a_{\text{min}}$ represents the largest and smallest values of the large sample data set. Hence, $a_{\text{max}} - a_{\text{min}}$ represents the range of the system.

*Chebyshev's inequality* states that 75% of the values in a normal dataset lie between 2 standard deviations of the mean. This is shown in Figure 5.1

![Chebyshev's inequality graph](image)

**Figure 5.1 Chebyshev's inequality graph**

Normalization in a statistical data is given by the formula $(X - \mu)/\text{S.D.}$ Assuming unit standard deviation, the standard score is given $X - \mu$. 
Let $m$ represent the mean of the $LongValue$ and $ShortValue$. ‘$m$’ can take any of the following conditions.

$ShortValue < LongValue$:  
This indicates that the patient’s recent health parameters are lower than normal. In this case, $m > ShortValue$ and $m < LongValue$. $m$ is normalized, yet lesser than the $LongValue$.

$ShortValue = LongValue$:  
In this case, $m = ShortValue = LongValue$ $m$ is already normalized.

$ShortValue > LongValue$:  
This condition implies that the patient’s recent health parameters are higher than normal. In this case $m < ShortValue$ and $m > LongValue$ $m$ is normalized, yet higher than the $LongValue$.

Thus,

$$(LongValue_i + ShortValue_i)/2 - ((a_{max} + a_{min})/2)$$

Represents the standardized normalization and gives the threshold to compute if the subsequent $ShortValue_i \sim LongValue_i$ belongs to the ‘normal’ or ‘abnormal’ data.

Hence, if $(ShortValue_i \sim LongValue_i) > DiffThresh$ there is a possibility of an abnormality.
Prediction

Prediction is done using the following formula.

\[ a_{i}^{t+1} = a_{i}^{t} \cdot \alpha + \text{historyFactor} \cdot a_{i}^{t}. \] (5.3)

\[ \alpha = \text{SensitivityRange}/\text{LongValue} \]

*SensitivityRange* represents the levels of sensitivity as decided by the doctor. The sensitivity value indicates whether the patient is in the Risk(High), Normal or the Risk (Low) range. For example in case of heart rate prediction *SensitivityRange* can be high (90), normal (85) or low (80). Now we have three cases

1. \( \alpha < 1 \) if \( \text{SensitivityRange} < \text{LongValue} \)
2. \( \alpha = 1 \) if \( \text{SensitivityRange} = \text{LongValue} \)
3. \( \alpha > 1 \) if \( \text{SensitivityRange} > \text{LongValue} \)

Case 1 represents the state where the patient is expected to produce a parametric value less than the *LongValue*

Case 2 represents a case where abnormality prediction does not depend on the *SensitivityRange* i.e, the patient is not sensitive to a particular parameter.

Case 3, represents the state when the patient is expected to produce an abnormality on the higher side.

\[ a_{i}^{t} \cdot \alpha \] represents the grouping of the next prediction, in order to group the value into one of the three levels.
e.g. \( a_i \), \( \alpha \approx 85 \) if and only if \( SensitivityRange \) is high

\( a_i \), \( \alpha \approx 80 \) if and only if \( SensitivityRange \) is normal

\( a_i \), \( \alpha \approx 75 \) if and only if \( SensitivityRange \) is low

The parameter \( historyFactor \) represents the contribution of the patient history in the prediction of the next value. The value of the \( historyFactor \) is determined by the doctor and it can take values from .01 to .05 (in steps of .01). The empirical \( historyFactor \) contributes to insinuate an abnormality i.e. if the patient has a history of heart problem, then the \( historyFactor \) will push the predicted value towards an abnormality.

Expression in (5.3) tends towards an abnormality if and only if \( \alpha \) tends towards an abnormality i.e. patient is expected to produce an abnormality.

\( \alpha \) tends towards an abnormality if and only if \( SensitivityRange \) tends towards an abnormality or if the difference between the \( longValue \) and \( SensitivityRange \) is too high.

5.3 RESULTS AND DISCUSSION

The proposed ODReL algorithm is validated using the heart rate data from the UCI machine learning repository for checking the abnormality detection and prediction of the subsequent values. It is found that the abnormalities are detected. The predicted values are compared with the actual value provided in the data set and it is observed that the proposed
algorithm predicts with a Root Mean Square Error (RMSE) of 1.8. RMSE is calculated using the formula:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}}$$

(5.4)

Where $y_t$ is the actual value.

$\hat{y}$ is the predicted value.

$n$ is the total number of values

Performance of the proposed ODReL algorithm is compared with the performance of Online Linear Regression for the same data set. Figure 5.2 shows the comparison of the error between the ODReL algorithm and the Online Linear Regression algorithm for the same data set. Error is calculated as follows:

Error = Actual Value - Predicted Value

(5.5)

The dotted line represents the expected error. It can be seen from the graph that the error for the ODReL reaches and almost coincides with the expected error as the time progresses. But Online Linear Regression produces higher errors as time progresses. Thus it is evident that ODReL performs better than OLR in the given data range.
Performance of the proposed ODReL algorithm is also compared with the Steepest Descent algorithm. Figure 5.3 shows the results of comparison. The green colored line shows the actual data. It can be observed from Figure 5.4 that the ODReL algorithm (in dotted lines) is able to predict the values accurately when compared to the Steepest Descent algorithm (Solid line), which could not produce consistent results for varied step sizes.
5.4 SUMMARY

This chapter proposed an Online Dynamic Linear Regressive Learning algorithm. The proposed ODReL algorithm learns the health parameters of a person in an incremental fashion. The algorithm is validated for the prediction of health parameters and abnormality detection in the health monitoring application. The algorithm detects the abnormality and predicts the next value of the parameters with commendable efficiency when compared with the state of the art algorithms like Online Linear Regression and Steepest Descent. Next instance prediction accuracy of the proposed algorithm can further be improved by taking the error difference between the predicted value and the actual value as the feed back.