

Title	Modeling meta-cognition for efficient learning
Author(s)	Ruan, Pingcheng
Citation	Ruan, P. (2014). Modeling meta-cognition for efficient learning. Student research paper, Nanyang Technological University.
Date	2014
URL	http://hdl.handle.net/10220/26034
Rights	© 2014 The Author(s).

Modeling Meta-Cognition for Efficient Learning

Ruan Pingcheng
School of Computer Engineering

Asst Prof Suresh Sundaram
School of Computer Engineering

Abstract - ‘Meta-cognitive Radial Basis Function Network’ (McRBFN) and ‘Projection Based Learning’ (PBL) is a machine-learning algorithm used to classify a data sample. Its meta-cognitive component selects one learning strategy from sample deletion, neuron growth and parameter update and sample reservation. The cognitive component adjusts the output weight to minimize the error of prediction using PBL algorithm. In this paper, we propose an improvement on the sample addition strategy in order to prevent the corruption of existing knowledge. At last, we evaluate the improved algorithm using three benchmarking classification problems.

Keywords - Machine-learning; Label Classification; Data-mining

1 INTRODUCTION

1.1 PROBLEM DEFINITION

$\{(\mathbf{x}^1, c^1), (\mathbf{x}^2, c^2) \dots (\mathbf{x}^t, c^t)\}$ is a sequence of data sets for learning. The input \mathbf{x}^t is a m-dimensional vector and its superscript denotes to be the t th sample. c^t is the class label of \mathbf{x}^t and the value of c^t is between 1 and n . The coded class label \mathbf{y}^t for c^t is defined as an n -dimensional vector based on the following function:

$$\mathbf{y}^t = [y_1^t \dots y_n^t]$$

$$y_j^t = \begin{cases} 1 & (c^t = j) \\ -1 & (c^t \neq j) \end{cases}$$

The objective of algorithm is to establish a single-hidden-layer neuron network (Figure 1) with suitable parameters to capture the relationship between \mathbf{x}^t and \mathbf{y}^t .

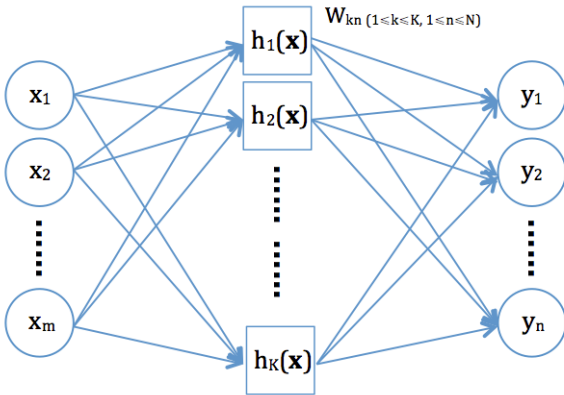


Figure 1 Neuron Network Model

In the neuron network, K refers to the current number of neurons and h_k^t represents the response of k th ($1 \leq k \leq K$) neuron to the t th sample. The extent of neuron's

activation is defined in the following function where u and σ is the center and width of k th neuron. Superscript l represents this neuron's class association.

$$h_k^t = \exp\left(-\frac{\|\mathbf{x}^t - \mathbf{u}_k^l\|^2}{(\sigma_k^l)^2}\right)$$

In this way, the predicted output $\hat{\mathbf{y}}^t$ for the t th sample can be computed using this formula where w_{kj} is the weight of k th neuron to the j th output.

$$\hat{y}_j^t = \sum_{k=1}^K w_{kj} h_k^t \quad j \in (1, n) \quad (1)$$

The predicted $\hat{\mathbf{c}}^t$ can then be computed in an inverse way.

$$\hat{c}^t = \arg \max_{j \in (1, n)} \hat{y}_j^t \quad (2)$$

In the next section, we introduce ‘Projection Based Learning’ (PBL) algorithm to adjust w_{kj} to minimize the error between \mathbf{y}^t and $\hat{\mathbf{y}}^t$.

1.2 PBL ALGORITHM [1]

The error of prediction for a given i th sample is measured by the energy function J_i .

$$J_i = \sum_{j=1}^n (y_j^i - \hat{y}_j^i)^2 = \sum_{j=1}^n (y_j^i - \sum_{k=1}^K w_{kj} h_k^i)^2$$

$$i \in (1, t) \quad j \in (1, n)$$

For all the t training samples, the sum of energy is a function of w_{kj} , or \mathbf{W} in Matrix Form.

$$J(\mathbf{W}) = \sum_{i=1}^t \sum_{j=1}^n (y_j^i - \sum_{k=1}^K w_{kj} h_k^i)^2$$

Taking the first derivative and equating it to zero, we get

$$\sum_{k=1}^K \sum_{i=1}^t h_k^i h_p^i w_{kj} = \sum_{i=1}^t h_p^i y_j^i$$

Substituting self-defined Matrix \mathbf{A} and \mathbf{B} to the above formula

$$\mathbf{A} = a_{kp} = \sum_{i=1}^t h_k^i h_p^i \quad (3)$$

$$\mathbf{B} = b_{kp} = \sum_{i=1}^t h_p^i y_j^i \quad (4)$$

Finally we get

$$W = A^{-1}B \quad (5)$$

1.3 PROBLEM AND IMPROVEMENT

In the meta-cognitive component, the response of current neuron network towards the new sample determines its learning strategy. The cognitive component uses to PBL algorithm to adjust the weights.

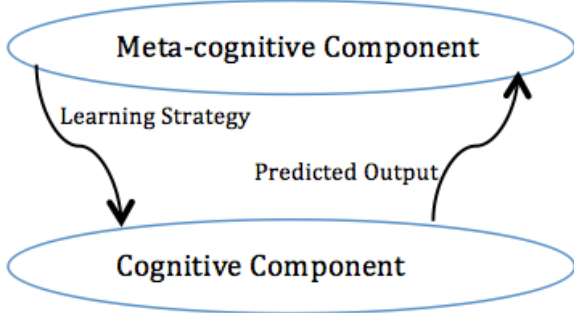


Figure 2 McRBFN-PBL Model

When meta-cognitive component chooses addition as the learning strategy for a new sample, the new neuron's parameters (u and σ) are only determined by the new sample without any information on the previous ones. Hence, the current neuron network may not predict well the proceeding samples as before. In this way, new knowledge may cause corruption of original knowledge.

To overcome the above-mentioned problem, we propose to use hidden neurons as pseudo-samples and reserved samples to measure the feeling of knowing. Then further action can be taken towards the new neuron according to this indicator.

In the experiment, we compare and analyze the testing accuracy with original algorithm versus improved one on three data sets, Image Segmentation, Iris and Wine to validate this improvement.

2 ALGORITHM

2.1 ORIGINAL ALGORITHM

In the original algorithm, the class label of new sample is firstly predicted using the current neuron network model. If the predicted output is the same as the actual output and the probability of right prediction is high enough, this sample will be deserted and will not be contributed to the learning process. If the predicted output is different from actual one or the error of prediction is larger than a threshold, the knowledge in this sample is fresh enough. Hence, a new neuron dedicated for this sample will be added to the current model. The parameters (u and σ) of this new neuron are determined by this sample and its relationship with other existing neurons. If the predicted and the actual output are the same but the error is larger than a

threshold, parameter update strategy will be applied to the neuron network and the weights will be adjusted based on the PBL algorithm. If this sample does not meet the above three criteria, sample reservation strategy will be applied and this sample will be added to the reserved queue for later inspection.

2.2 PROCEDURE OF IMPROVEMENT

In the neuron growth strategy, before a new neuron is added to the model, we keep a snapshot of original model. Then we do the following two operations to validate the new network model after the neuron's addition.

First of all, as the centers of hidden neurons records down the knowledge of past samples, we use hidden neurons as the pseudo-samples to check whether the new model corrupt the existing knowledge. If there exists an old neuron that the predicted label under the new model is different from its associated class, then the width (σ) of new neuron will be shrunk by 10 percent to reduce the effect of this neuron. Repeat the proceeding step for several times until no misclassified hidden neuron is found. If the iteration is more than ten times, the addition of this new neuron should be roll backed and assign the snapshot of original model back.

Second operation checks the previous reserved samples. If a previous correctly classified reserved sample is predicted incorrect under the current model, then this reserved sample will be used to adjust the weights and update the current model.

2.3 PSEUDOCODE

```

Initialize the number of neurons  $K = 1$ ;
Initialize the first neuron  $h_1$  according to the first
sample  $x_{i1}$ ;
Initialize Matrix  $A$  to be  $[1]$ ;
Initialize Matrix  $B$  to be  $[y_1]$ ;
Initialize an empty matrix for  $W$ ;

While the data stream is not empty:
    Pop the sample  $x^i$  at the head of data stream;
    Compute predicted output  $\hat{y}^i$  and expected label
     $\hat{c}^i$  using Equation (1) (2);
    Compute the maximum hinge error  $E^i = \max(|y^i - \hat{y}^i|)$ ;
    If  $c^i == \hat{c}^i$  and  $E^i$  is smaller than deletion
    threshold
        Desert this sample;
    Elseif  $c^i != \hat{c}^i$  or  $E^i$  is greater than addition
    threshold
        Keep a snapshot of original Matrix  $A$ ,  $B$  and
         $W$ ;
         $K = K + 1$ ;
        Add a neuron to the model based on the
        sample and its relationship with other
        neurons;
        Augment and update Matrix  $A$  and  $B$  based
        on Equation (3) (4);
        Compute Matrix  $W$  using PBL algorithm
        based on Equation (5);
        While there exists misclassified hidden
        neuron:
            If the iteration is more than 10 times:
                Remove the new neuron from
                model;
                Assign the snapshot Matrix  $A$ ,  $B$ 
                and  $W$  back;
                Break the loop;
            Else:
                Shrink the width of new neuron by
                10 percent;
            End if;
        End while;
        While there exists misclassified reserved
        samples which is correctly predicted
        previously:
            Update the model and adjust the
            weights according to this reserved
            sample;
            Desert this sample from Reserved
            Queue;
        End while;
    Elseif  $c^i == \hat{c}^i$  and  $E^i$  is greater than update
    threshold:
        Augment and update Matrix  $A$  and  $B$  based
        on Equation (3) (4);
        Compute Matrix  $W$  using PBL algorithm
        based on Equation (5);
    Else:
        Push the sample to the end of data stream for
        later learning;
        Set its flag if it is correctly predicted;
    End if;
End the learning;

```

3 EXPERIMENTS

3.1 IMAGE SEGMENTATION

Image Segmentation dataset [2] are collected from UCI Machine Learning Repository. Instances of data are drawn from one of seven outdoor images. The parameters of data sets are listed in the following table.

Table 1 Parameters of Image Segmentation

Number of Features (m)	Number of Classes (n)	Number of training instances	Number of testing instances
19	7	210	2100

The training and testing accuracy of the improved algorithm versus the original one are compared in the following table.

Table 2 Comparisons on Image Segmentation Samples

	Overall Training Accuracy	Average Training Accuracy	Overall Testing Accuracy	Average Testing Accuracy
Original Algorithm	98.57%	98.57%	92.05%	92.05%
Improved Algorithm	98.57%	98.57%	93.52%	93.52%

3.2 IRIS

Iris data set [3] includes 3 classes of Iris Plant, each of which contains 50 instances. The following table shows its relevant parameters.

Table 3 Parameters of Iris

Number of Features (m)	Number of Classes (n)	Number of training instances	Number of testing instances
4	3	45	105

Table 4 displays its relevant training and testing accuracy.

Table 4 Comparisons on Iris

	Overall Training Accuracy	Average Training Accuracy	Overall Testing Accuracy	Average Testing Accuracy
Original Algorithm	97.778%	97.778%	96.19%	96.19%
Improved Algorithm	100%	100%	96.19%	96.19%

3.3 WINE

Samples in Wine [4] data set are derived from chemical analysis of wines grown by 3 different cultivars. Its parameters are displayed as the following.

Table 5 Parameters of Wine

Number of features (m)	Number of Classes (n)	Number of training instances	Number of testing instances
13	3	60	118

Table 6 shows the prediction statistics on Wine data samples.

Table 6 Comparisons on Wine Samples

	Overall Training Accuracy	Average Training Accuracy	Overall Testing Accuracy	Average Testing Accuracy
Original Algorithm	100%	100%	97.458%	98.039%
Improved Algorithm	100%	100%	98.305%	98.693%

3.4 GRAPH

Figure 3 compiles all the prediction results of three data samples into one bar chart. From the bar chart, it is evident that the edited algorithm improves both the training and testing accuracy.

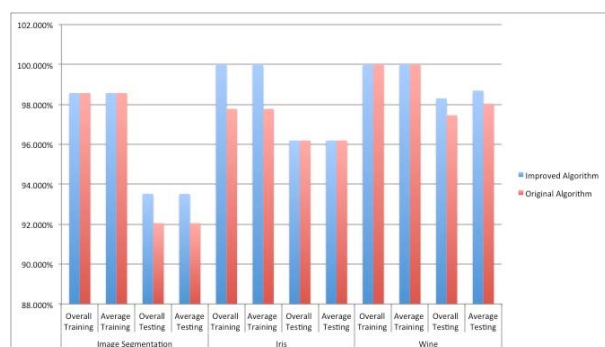


Figure 3 Prediction Results from Three Data Sets

4 CONCLUSION

In this paper, we have presented an improvement on the McRBFN-PBL algorithm. In the original algorithm, the addition of new neuron towards the current model may degrade model's performance on previous samples and corrupt the existing knowledge. We address this problem by examining the new model on hidden neurons and reserved samples. We use both of them as representatives of the past samples and readjust the model for better performance. At last, we evaluate the improved algorithm on UCI machine learning repository and compare the prediction results with the original one. The statistics shows the superior performance of our improvement.

ACKNOWLEDGMENT

I would like to express the depth of my gratitude to Assistant Professor Suresh Sundaram for his patience and guidance for my research work. My achievement is inseparable from his constant instructions. I would also

like to thank Mr Rangarajan Badrinarayanan for his invaluable help on solving my doubt.

We wish to acknowledge the funding support for this project from Nanyang Technological University under the Undergraduate Research Experience on Campus (URECA) programme.

REFERENCES

- [1] G.Sateesh Babu, and S.Suresh, "Meta-cognitive RBF Network and its Projection Based Learning algorithm for classification problems", *Applied Soft Computing*, vol. 13, pp. 654-666, 2013.
- [2] Vision Group, University of Massachusetts. (1990, Nov 1) . *Statlog (Image Segmentation) Data Set [Online]*. Available: <http://archive.ics.uci.edu/ml/datasets/Statlog+%28Image+Segmentation%29>
- [3] R.A. Fisher. (1988, July 1). *Iris Data Set [Online]*. Available: <http://archive.ics.uci.edu/ml/datasets/Iris>
- [4] Forina, M. et al, (1991, July 1) *Wine Data Set [Online]*. Available: <https://archive.ics.uci.edu/ml/datasets/Wine>