UPDATING STRATEGIES FOR DISTANCE BASED CLASSIFICATION MODEL WITH RECURSIVE LEAST SQUARES

Anna-Maria Raita-Hakola[a,*], Ilkka Pölönen[a],

[a] Faculty of Information Technology, University of Jyväskylä, 40100, Jyväskylä, Finland -
(anna.m.hakola, ilkka.polonen)@jyu.fi

Commission III, WG III/4

KEY WORDS: Hyperspectral imaging, Minimal Learning Machine, Recursive Least Squares, Classification, Real-time computation, Machine learning

ABSTRACT:

The idea is to create a self-learning Minimal Learning Machine (MLM) model that is computationally efficient, easy to implement and performs with high accuracy. The study has two hypotheses. Experiment *A* examines the possibilities of introducing new classes with Recursive Least Squares (RLS) updates for the pre-trained self learning-MLM model. The idea of experiment *B* is to simulate the push broom spectral imagers working principles, update and test the model based on a stream of pixel spectrum lines on a continuous scanning process. Experiment *B* aims to train the model with a significantly small amount of labelled reference points and update it continuously with (RLS) to reach maximum classification accuracy quickly.

The results show that the new self-learning MLM method can classify new classes with RLS update but with a cost of decreasing accuracy. With a larger amount of reference points, one class can be introduced with reasonable accuracy. The results of experiment *B* indicate that self-learning MLM can be trained with a few reference points, and the self-learning model quickly reaches accuracy results comparable with nearest-neighbour NN-MLM. It seems that the self-learning MLM could be a comparable machine learning method for the application of hyperspectral imaging and remote sensing.

1. INTRODUCTION

For remote sensing and Earth observation, hyperspectral imaging (HS Imaging) provides powerful tools for a variety of interesting applications, such as monitoring vegetation, environmental parameters, agricultural and forestry phenomena (Adão et al., 2017, Tuominen et al., 2018, Honkavaara et al., 2013, Nevalainen et al., 2017, Nezami et al., 2020). Due to the continuous HS imaging technology development, the sizes of the HS imagers has been decreasing, and the HS imaging sensors have become more affordable. So far, small and lightweight HS imagers can be applied to unmanned aerial systems (UAS), which are popular and cost-effective platforms in remote sensing (Adão et al., 2017).

The HS data is highly dimensional, offering high accuracy and robustness for identification and characterisation tasks (Camps-Valls and Bruzzone, 2005, Bioucas-Dias et al., 2013). The strengths and weaknesses of HS imaging in remote sensing applications arise from the essence of the data. Each (HS) image can be considered as a stack of frames, each representing an intensity of a different wavelength of light. The data contains spatial and spectral information, and each pixel has its spectrum, which is a strength compared to other less informative imaging methods. The downside of remote sensing HS imaging applications is a large amount of data due to the spectral channels. Other challenges can be the platform's payload and processing constraints, restricted available energy and the complexity of the machine learning methods (Haut et al., 2018, Caba et al., 2020). Thus, computationally efficient, easy to implement and adaptive machine learning methods are needed.

Minimal Learning Machine (MLM) is a distance-based supervised method for classification, regression and anomaly detection, which performance is comparable to most state-of-the-art methods (Mesquita et al., 2017, Pölönen et al., 2020, Raita-Hakola and Pölönen, 2021, Hakola and Pölönen, 2020). MLM is easy to implement. Its learning phase consists of building a linear mapping between input and output distance matrices. MLM uses this model in the generalisation phase for estimating the distances from the new output reference points to the target output values. In classification, the output estimation can be seen as an optimisation problem or, we can use the MLM Nearest neighbour (NN-MLM) variation, a computationally efficient alternative. The effiency is reached by using a simple linear search over a set of distance estimations (Mesquita et al., 2017).

Since the core of this method is the linear mapping, a model that can be solved with the Ordinary Least Squares estimator (OLS), it is an interesting idea to study if it is possible to create a new variation from the MLM, which updates itself with Recursive Least Squares (RLS) algorithm. This study examines the MLM with RLS, aiming to test two hypotheses.

Hypothesis *A* is that we can train new classes to the MLM model with new data and RLS algorithm without re-training the whole model. Since the MLM needs the whole training set X only at the beginning and operates with the significantly smaller subset R. Thus, it is interesting to increase the amounts of classes by introducing the new classes to the model with a new set of training points. The new training points are only a portion of the subset R, which ensures the model is computationally efficient.

Hypothesis *B*: it is possible to increase the accuracy of the MLM classifier by updating the pre-trained model with new information and a recursive least squares estimator. We can use only a few reference points for the first-round classifier and increase the accuracy with the model updates. This way, we could create a model that could learn from the new data and adapt the classifier better for changing targets, such as forest types or land covers.

*Corresponding author.

The hypothesis is that the MLM model can be updated without re-training the whole model, and the new self-learning MLM can adapt to its new data for increased accuracy. This kind of machine learning method could be suitable, for example, for real-time push broom HS imaging applications. In this study, we call this new approach to self-learning MLM.

The paper is organised as follows. Section 2. describes the methods and the selected hyperspectral data. The results are introduced in section 3.. The discussion is the fourth section, and the final conclusions are in the section 5..

2. MATERIAL AND METHODS

2.1 Minimal Learning Machine

The Minimal Learning Machine offers computationally efficient solutions for classification and anomaly detection (Pölönen et al., 2020, Raita-Hakola and Pölönen, 2021) applications. It is a computationally efficient and fast machine learning method with the basic idea of utilising the mapping between the input and output distances (de Souza et al., 2015).

Let $X \subset \mathbb{R}^d$ be a training set consisting from spectra and $\mathbf{x}_i \in X$ a single spectrum i. Also, let $R \subset X$ be a set of reference points and $\mathbf{m}_k \in R$ a single spectrum k. Corresponding, the training set ground truth labels are $\mathbf{y}_i \in Y \subset \mathbb{R}$ and for the subset of Y, labels $\mathbf{t}_k \in T$. The training set X has N samples, and the subset R size is K. Now, the distances for the linear mapping are $d(\mathbf{x}_i, \mathbf{m}_k)$ and $\delta(\mathbf{y}_i, \mathbf{t}_k)$.

When T and R are selected, we can define two distance matrices $\mathbf{\Delta}_y \in \mathbb{R}^{N \times K}$ and $\mathbf{D}_x \in \mathbb{R}^{N \times K}$ using previous mappings d and δ. By assuming the linear mapping between these two distance matrices, we have a linear model:

$$\mathbf{\Delta}_y = \mathbf{D}_x\mathbf{B} + \mathbf{E}, \tag{1}$$

where $\mathbf{B}$ is coefficients and $\mathbf{E}$ is the residual. We can approximate the coefficients $\mathbf{B}$ with the Ordinary Least Squares estimator (OLS) (Mesquita et al., 2017)

$$\widehat{\mathbf{B}} = (\mathbf{D}_x^T\mathbf{D}_x)^{-1}\mathbf{D}_x^T\mathbf{\Delta}_y. \tag{2}$$

As a result, we have a $\hat{\mathbf{B}}$, which is the linear model between the input distances $d(\mathbf{x}_i, \mathbf{m}_k)$ and output distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$.

Now, for the new spectrum $\mathbf{x}_n$ the distance between its label y_n and set T is

$$\delta(y_n, T) = d(\mathbf{x}_n, R)\widehat{\mathbf{B}}. \tag{3}$$

Label y_n can be estimated by solving an quadratic optimisation problem

$$\min_{y_n} \sum_{k=1}^{K} \Big(y_n - \mathbf{t}_k)^T (y_n - \mathbf{t}_k) - \delta^2(y_n, T) \Big)^2 . \tag{4}$$

The classification (equation 4) can be seen as multilateration problem, where quadratic optimisation can be used (de Souza et al., 2015). It is also possible to use Tikhonov regularisation to produce more robust models (Kärkkäinen, 2019). Besides the optimisation and regularisation possibilities, we can utilise the distances and consider the model $\widehat{\mathbf{B}}$ to be a generalisation of the nearest neighbour classifier (NN-MLM) (Mesquita et al., 2017). Equation 3 gives us distances, and for classifying the $\mathbf{x}_n$, we perform an argument sorting for $\delta(\mathbf{y}_n, T)$ and assign the N-closest label value from T. The detailed implementation of the nearest neighbour generalisation is in (Hakola and Pölönen, 2020).

2.2 Recursive least squares approach

Ordinary Least Squares (OLS), used in the MLM (equation 2), is meant for static situations. However, the OLS can be updated continuously with new data, using a Recursive Least Squares (RLS) algorithm. The updating is computationally efficient because it does not need the original data. The RLS is based on a matrix inversion lemma called the Sherman-Morrison-Woodbury equation (Meyer, 2000). According to this lemma, it is possible to update a linear equation system solution efficiently. Detailed proof and equations can be found from (Romberg, 2016, Haykin, 2008). Algorithm 1 describes the RLS updating phases, which are implemented and utilised in our following self-learning MLM model experiments.

Algorithm 1: Recursive Least Squares (Romberg, 2016)

Input: X,R,Y,T
Initialize:
Calculate distance matrix $\mathbf{D}_{x,0}$
Calculate distance matrix $\mathbf{\Delta}_{y,0}$
Calculate $\mathbf{P}_0 = (\mathbf{D}_{x,0}^T\mathbf{D}_{x,0})^{-1}$
Calculate model $\widehat{\mathbf{B}}_{x,0} = \mathbf{P}_0\mathbf{D}_{x,0}^T\mathbf{\Delta}_{y,0}$
for *i = 1,2,3...* **do**
  New data X_i, Y_i appears
  Calculate distance matrix $\mathbf{D}_{x,i}$
  Calculate distance matrix $\mathbf{\Delta}_{y,i}$
  Calculate $\mathbf{P}_{x,i} = \mathbf{P}_{i-1}\mathbf{D}_{x,i}^T$
  Calculate $\mathbf{P}_i =$
    $\mathbf{P}_{i-1} - \mathbf{P}_{x,i}(\mathbf{I} + \mathbf{D}_{x,i}\mathbf{P}_{x,i})^{-1}\mathbf{D}_{x,i}\mathbf{P}_{i-1}$
  Calculate $\mathrm{K}_i = \mathbf{P}_i\mathbf{D}_{x,i}^T$
  Update model $\widehat{\mathbf{B}}_{x,i} =$
    $\widehat{\mathbf{B}}_{x,i-1} + \mathbf{K}_i(\mathbf{\Delta}_{y,i} - \mathbf{D}_{x,i}\widehat{\mathbf{B}}_{x,i-1})$
end for

2.2.1 Self-learning MLM: Experiment *A*

In experiment *A*, we trained and updated the model in phases. The aim was to train the model only with a few classes and then use RLS updates to introduce new classes.

At first, we selected an equal amount (*class-wise R size*) of data points to R and their labels T from each of the first four classes. The first model was trained on the initialization phase in Algorithm 1.

After the initialization, the Algorithm 1 got new data X_i, Y_i. The size of the new data was the *class-wise R*, and it contained samples only from a new class. The model was updated and tested with test data containing the previous and new data class samples. The updated model provided the new distances $\delta(\mathbf{y}_n, T)$. Since the T contains only samples from classes one to four, we needed to alter how we selected new classes. At first, we classified the data with T, and nearest neighbour method (Hakola and Pölönen, 2020). Classes from one to four got their labels.

Then we used classification rules shown in Fig. 1. As we know the new datapoint distances to the reference data points $\delta(\mathbf{y}_n, T)$, we can re-check the predictions that had the biggest class label four (4). The distance from a new label to the first label is longer than the longest distance of the last known biggest label four (4). We can see from Fig. 1 that the Euclidean distance from label four (4) to label four (4) is 0. The longest known distance from label four (4) to label one (1) is 3. Therefore, the distance from a new label five (5) to label one (1) is more than 3.5.

Based on the known distances, the halfway limit was set to 0.5. With this limit, we used a majority voting rule: if more than

80% (calculated from the *class-wise* R size) of the distances from new data point to reference points are longer than the last round's *biggest class* -0.5, the point is classified to a new class, which is the last round's *biggest class* + 1. For every new class, we computed the classes similarly, updating the *biggest class* in a loop, until all of the new classes were re-classified.

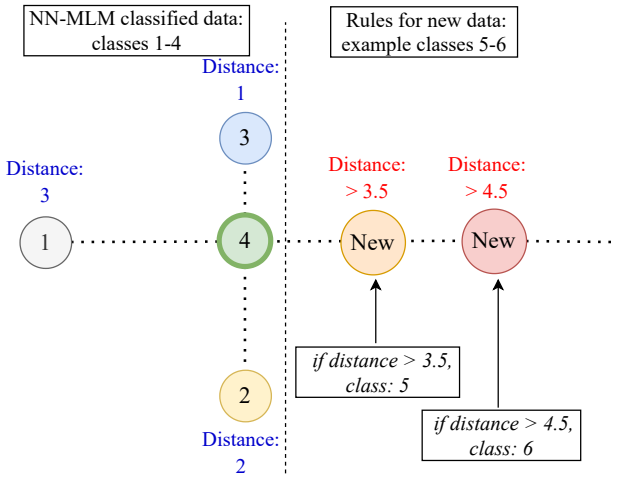

Figure 1: Classification rules for the new classes. Classes 1-4 were classified using the NN-MLM. The distances from the biggest known class four (4) are written in blue. The image show the first updating rules. The longest distance of the class four (4) is 3, all distances longer than 3.5 were re-classified to a bigger class. The biggest class was updated and again, the longest distance was set to 4.5 (5 - 0.5), and the data was re-checked. Each time a new class appeared, all of the biggest class datapoints were re-checked with this halfway limit as many times as we had new classes.

2.2.2 Self-learning MLM: Experiment *B* This experiment *B* aims to simulate the push broom spectral imagers working principles, update and test the model based on a stream of pixel spectrum lines on a continuous scanning process. We will limit the amount of the reference points R and the sizes of the distance matrices $\mathbf{D}_x$ and $\mathbf{\Delta}_y$, train the first model and update it with new data and its new ground truth, labelled with the previous model, row-by-row.

Usually, machine learning methods need a lot of training data. MLM differs, it is a method that uses only a subset (R) of the original data X, and therefore it is computationally efficient. The first model can be trained with only a few known labelled pixel spectra in this experiment. Instead of using the X and Y in distance calculations, the first model was trained (Algorithm 1, initialize) using only the distances between R to R and T to T. This reduces the number of distance calculations and decrease the distance matrices sizes.

The first model was given to the RLS update loop. There were 41 rows in the training and testing datasets. For each row (image A, Fig. 4), the model was updated and tested. Each time the given new data $\mathbb{X}_i$ got an updated ground truth $\mathbb{Y}_i$. For the updated ground truth, we classified the new training points $\mathbb{X}_i$ with a previous model $\widehat{\mathbf{B}}_{i-1}$. As a result, the model's behaviour is self-learning since each time it updates itself, it specifies and updates its ground truth knowledge based on the current update and uses it for the next model update, reaching for increasing accuracy.

We run tests for each row. In row-by-row testing (RBR), the test data $\mathbb{X}_{test,_i}$, $\mathbb{Y}_{test,_i}$ was a new row, top-down from the visualised image B Fig. 4. Besides the RBR testing, each updated model was tested with whole test data (TDR testing). The average number of pixels (each with 103 spectral channels) in an RBR's training and test row was 67 since the unlabelled pixels were removed from the data.

Experiments were implemented using Scikit-Learn (Pedregosa et al., 2011) and SciPy (Virtanen et al., 2020) Python libraries. The computing was performed with 28 core Linux server for non-parallel computing, x86_64.

2.3 Hyperspectral data

The experiments were done with two known hyperspectral images, Pavia University and Salinas A scene. The data is provided by the Grupo De Intelligencia Computational (GIC)(Graña et al., 2022). The Pavia Centre HS image is acquired by the ROSIS sensor with 103 spectral channels with the spatial resolution of 1.3 meters and with a spectrum coverage ranging from 430 to 860 nm (Xie et al., 2019). The image size is 1096 x 1096 pixels. The image ground truth (Fig. 2) is divided into 9 classes (Graña et al., 2022).

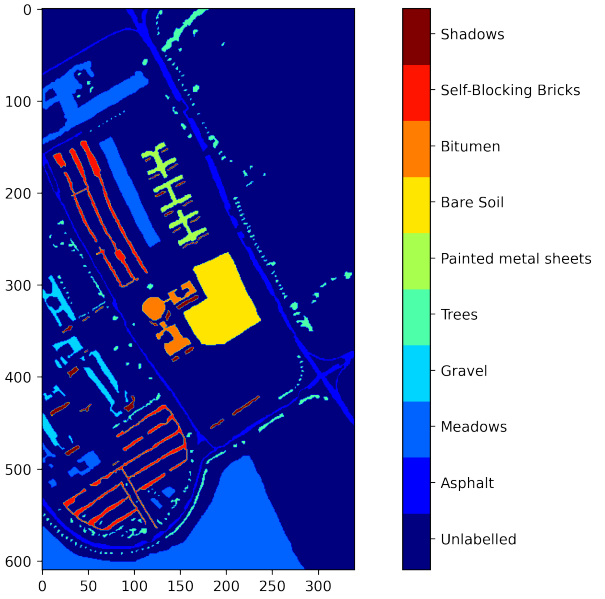

Figure 2: Visualization of the Pavia University ground truth. The image size is 610 x 340 pixels with 103 spectral channels.

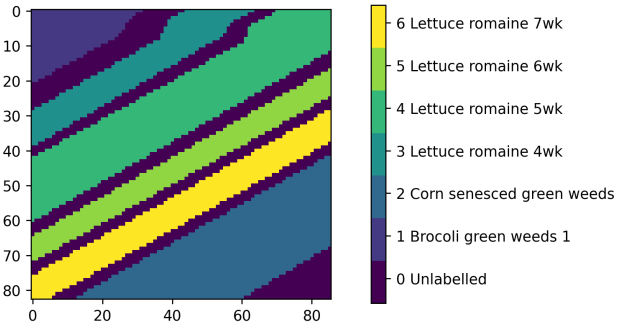

Figure 3: Visualization of the Salinas A scene ground truth. The image size is 83 x 86 pixels with 204 spectral channels.

We used the Salinas A scene collected with an AVIRIS sensor in the self-learning MLM *B* experiment. The image size is 83 x 86 pixels, it has 204 spectral bands, and the labels are divided into six classes (Fig. 3). The spatial resolution is 3.7 m over the range of 400–2500 nm (Sawant and Manoharan, 2020).

Both datasets were normalised between 0 and 1, and the unlabelled pixels were removed from the training and testing data. After the preprocessing, the datasets were divided into training and testing portions as follows. The detailed information of these datasets can be found from (Graña et al., 2022).

2.3.1 Pavia University, experiment *A* The Pavia dataset was divided class-wise with 50:50 portions to training and test datasets. After the split, the data was randomised. The idea of this dividing- and pre-processing logic enabled was the introduction of the new classes for the self-learning MLM model. This way, we could easily train the first model with four first classes and test it with four class customised test data. During the next rounds, we could introduce only new class data points from the training data for the model and test the updated model with a new customised test set, including the old and new classes. After removing the unlabelled data, the training data size was 21391 and the test data 21385 pixels with 103 spectral channels.

2.3.2 Salinas A scene, experiment *B* The Salinas dataset was divided horizontally top-down with portions of 50:50. Every other row to test data, every other to training data. Fig. 4 visualises the training (A) and test data (B) on frame 144, representing wavelength 1890 nm. After removing the unlabelled data, the training data size was 2707 and the test data 2641 pixels with 204 spectral channels.

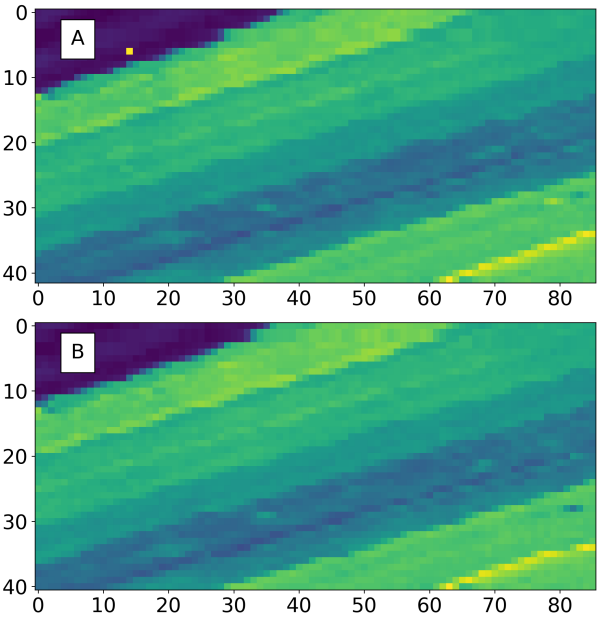

Figure 4: Salinas test and training data. Images A and B represent Salina's frame number 144 (wavelength 1890 nm) after the slicing. The dataset was divided horizontally, top-down. Every other row to training data (A), every other to test data (B).

After selecting the equal amount of the reference points from the randomised test data class-wise, the self-learning MLM model utilised the training data row by row, top-down. The accuracy of the model was tested row-by-row (RBR) with test row (next index from the original data) and with the whole test data (TDR).

3. RESULTS

3.1 Experiment *A*

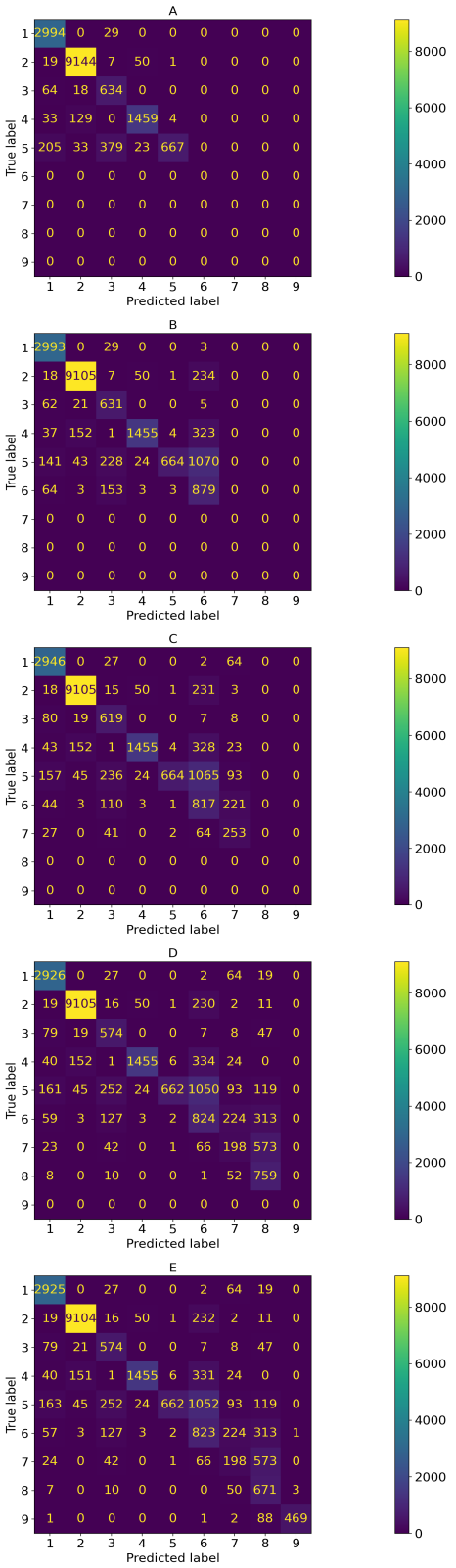

Figure 5: Experiment *A* confusion matrices. The images visualise the classification results after the new classes are introduced to the self-learning MLM. Image A new class: 5, B new class: 6, C new class: 7, D: new class 8, E: new class 9.

In experiment *A*, we introduced new classes to the self-learning MLM. Fig. 5 visualises the confusion matrices of the classification results. Images A-E show (Fig. 5) how the self-learning MLM classifies the first model's classes (1-4) and the rest of classes 5-9. Image A has classes 1-5, B 1-6 until finally, image E has all of the Pavia dataset's nine classes.

The self-learning MLM classifier can recognise and classify new classes with RLS updates and classification rules from Fig. 1, but with a cost of decreasing accuracy. Table 1 show that the, and the R size affects the results, increasing the accuracy, but the newly introduced classes are decreasing it.

Class-wise R size	5	6	7	8	9
100	0.90	0.80	0.78	0.74	0.73
200	0.92	0.82	0.80	0.76	0.76
500	0.94	0.85	0.83	0.79	0.79
800	0.94	0.86	0.84	0.80	0.80
1000	0.94	0.87	0.85	0.80	0.80

Table 1: Experiment *A* accuracy results for each new class 5-9. Class R size: number of training points selected per class.

3.2 Experiment *B*

Experiment *B* results in table 2 show that the self-learning MLM is reaching comparable accuracy against the once trained MLM (OTM).

Model	Class R, N.N.	Max	Min	Mean	Mode
RBR	10,3	1.00	0.86	0.97	1.00
TDR	10,3	0.97	0.95	0.96	0.96
OTM	10,3	0.98	0.98	0.98	0.98
RBR	10,5	1.00	0.86	0.97	1.00
TDR	10,5	0.97	0.95	0.96	0.96
OTM	10,5	0.98	0.98	0.98	0.98
RBR	10,10	1.0	0.86	0.97	1.00
TDR	10,10	0.97	0.95	0.96	0.96
OTM	10,10	0.98	0.98	0.98	0.98
RBR	20,3	1.00	0.89	0.98	1.00
TDR	20,3	0.98	0.97	0.98	0.98
OTM	20,3	0.99	0.99	0.99	0.99
RBR	20,5	1.00	0.89	0.98	1.00
TDR	20,5	0.98	0.97	0.98	0.98
OTM	20,5	0.99	0.99	0.99	0.99
RBR	20,10	1.0	0.89	0.98	1.00
TDR	20,10	0.98	0.97	0.98	0.98
OTM	20,10	0.99	0.99	0.99	0.99
RBR	100,3	1.00	0.94	0.99	1.00
TDR	100,3	0.99	0.99	0.99	0.99
OTM	100,3	0.99	0.99	0.99	0.99
RBR	100,5	1.00	0.94	0.99	1.00
TDR	100,5	0.99	0.99	0.99	0.99
OTM	100,5	0.99	0.99	0.99	0.99
RBR	100,10	1.0	0.94	0.99	1.00
TDR	100,10	0.99	0.99	0.99	0.99
OTM	100,10	0.99	0.99	0.99	0.99

Table 2: Maximum, minimum, mean and mode values of the accuracy. RBR: row-by-row updated model, one-row test data accuracy, TDR: row-by-row updated model, whole test data row-by-row model accuracy, OTM: once trained MLM model. Class R: number of selected training reference points per class, N.N.: number of neighbours.

The row-by-row results (RBR) in Table 2 and Fig. 6 has the most variation. The single RBR row accuracy result might be weak, but

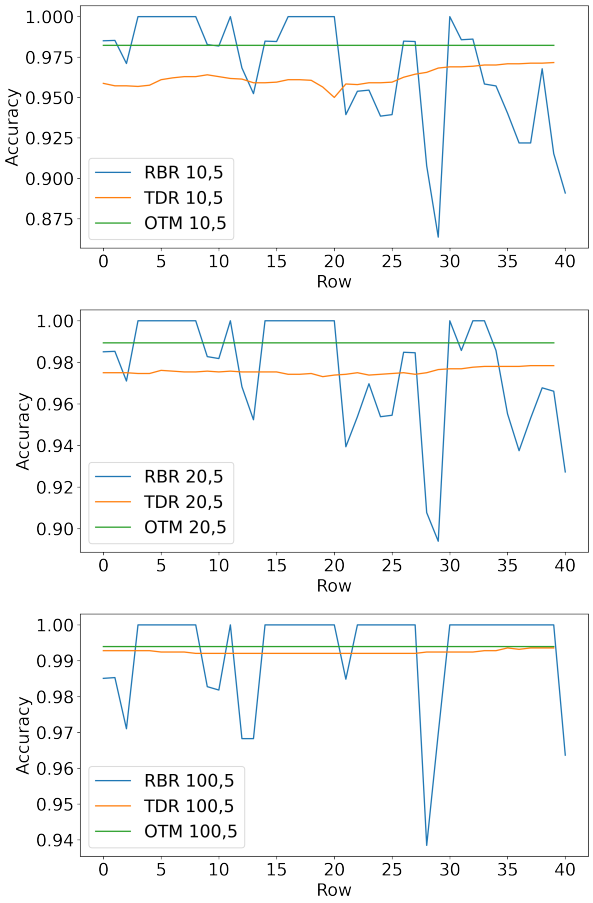

Figure 6: Visualisation of the self-learning MLM accuracy results. RBR: self-learning MLM, one-row accuracy, test data is original image's next row from the current update row. TDR: self-learning MLM, the accuracy of the whole classified test data, measured after very row-by-row model updates. OTM: once trained MLM model, tested with the whole test data. First image: *classwise R size*: 10, number of neighbours: 5, second image *classwise R size*: 20, number of neighbours: 5, third image *classwise R size*: 100, number of neighbours: 5.

the overall effect of the single row low performance is reasonable for the whole model. For example, the self-learning MLM model that was trained with 10 as a *classwise R size* had the lowest single row accuracy of 86%. The same model had a mean accuracy of 97%, and the mode of the one-row pixel-wise classification accuracy results was 100%. The model was also tested row-by-row with the whole test data (TDR), reaching the mean accuracy of 96%.

In Fig. 6, we can see the accuracy results row by row. RBR, TDR (self-learning model, tested after every row with whole test data), and OTM (once trained MLM) models had the same *classwise R size* and the number of neighbours. The effect of the increasing accuracy of the *classwise R size* for the self-learning MLM can be seen in Fig. 8, which visualises the accuracy results calculated row-by-row from the classified whole test data. The number of neighbours seemed to be a parameter that did not affect the results of this experiment.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-3-2022-163-2022 | © Author(s) 2022. CC BY 4.0 License.

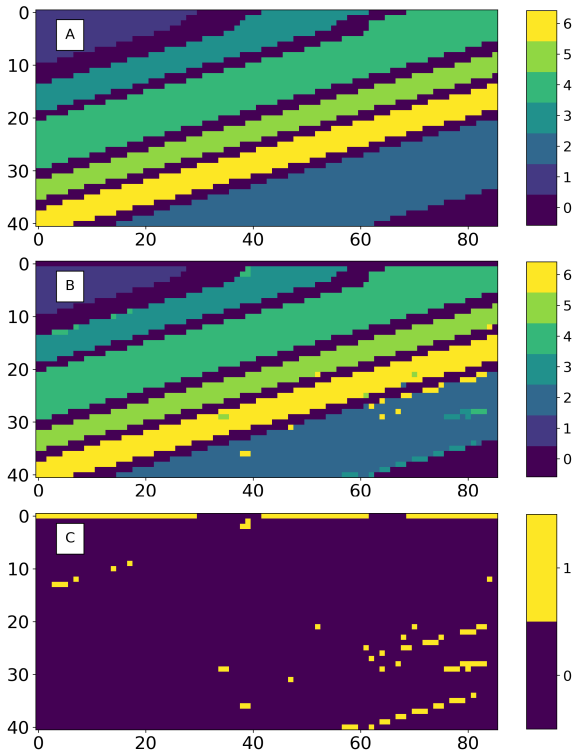

Figure 7: A: Salinas test data ground truth, B: Self-learning MLM's top-down, row by row (RBR) classification map, C: Row-by-row classification error map.

Figure 8: Self-learning MLM accuracy results row-by-row, using each time the whole test data.

Self-learning MLM's row-by-row classification results can be seen on the classification map (Fig. 7). Image A is the ground truth of the test data. B visualises the results that are classified row-by-row from top to down. Between every row shown in the image, the model is updated with one row of new training data (training and test data are visualised in Fig. 4. The labels for the new training data are produced with the previous updated model. Image C in Fig. 7 shows the classification errors of the RBR results. Yellow represents miss-classification; purple is correct.

4. DISCUSSION

The results of the self-learning MLM experiment *A* show that the model can handle new classes after the RLS updates, but the accuracy decreases with every appearing new class. Similarly, with the previous research, when the size of R approaches the size of X, the accuracy of the results improves (de Souza et al., 2015). The number of samples in the Pavia dataset was not classwise equal; for example, class seven (7) had only 476 samples, which might have affected the results since the maximum of the updating phase's *classwise R size* in that class was 238.

In our experiment, the number of the training samples in the reference set R had a positive effect on the results, which means that with a larger amount of training data, the first new class can be updated to the model with reasonable accuracy (Table 1, Fig. 5).

However, for introducing several new classes, we would recommend either improving the classification method 1, using the optimization methods (de Souza et al., 2015, Kärkkäinen, 2019) or considering training a new model with accurate ground truth after every n:th update rounds. Further research is needed for increasing the accuracy of introducing multiple new classes to the self-learning MLM.

Self-learning MLM experiment *B* seemed to reach our aim of creating a method for a model that can be trained with a low number of samples, and it will adapt itself to the data stream. Originally, the MLM training distances were the distances between the training set X, subset R, and the corresponding label distances Y and T (de Souza et al., 2015). Our once trained model (OTM) calculated distances from each point of the R to each point of the X. The self learning-MLM calculated the input distances from R to R and output distances from T to T.

As an example, we had six (6) classes, and the *classwise R size* was 20. OTM calculated the distances from 120 reference points to all 2707 data points (and similarly the label distances). Our approach calculated the distances using only the reference points R and their labels T, which means distances from the 120 points to 120 data points and similarly with the labels. As a starting point, the first model can be trained with 20 or 100 pixel spectra per class, and it can reach the accuracy of the OTM model (98-99%) by updating and learning from the incoming data stream. From the computational performance perspective, since we calculate the distances row-by-row, the self-learning MLM is close to the NN-MLM classifier even though the starting point is less demanding.

One of the MLM's benefits against many other machine learning models is the low number of hyperparameters. A comparable NN-MLM can be trained, depending on the classification task, with three hyperparameters: the number of neighbours, the R size and distance calculation method (Hakola and Pölönen, 2020). We selected the Euclidian distance method based on the previous study of MLM classifier, and hyperspectral data (Hakola and Pölönen, 2020). In that study, the number of neighbours affected the accuracy. Self-learning MLM seems to be more ignorant on

the number of neighbours. The most effective hyperparameter was the *classwise R size*. In experiment *A*, the R size had a more significant role on the results, where the experiment *B* (Table 2) show that the self-learning MLM can reach high accuracy without a large number of training data, using only few class-wise spectra as a reference data.

The row-by-row approach was semi-sensitive to the quality of the incoming data. Fig. 6 shows how the RBR accuracy varies depending on the single training or testing row of pixel spectra. This variation seems to be a phenomenon with only a small effect on the results conducted using the same row-by-row models with the whole test data between the updating and learning processes.

RBR classification error map (Fig. 7, image C) reveals the missclassified pixels. Visual evaluation over the spectral channels of the training and test data shows that there might be some anomalous pixel spectra that differ from the other similarly annotated data around them. For example, the yellow line on the right bottom corner and blue dots on the green area on top of it in image B (Fig. 4) can be seen on spectral channel 1890 nm. Similar features can be seen on the classification error map but not in the ground truth image (Fig. 3). That is an interesting point of view for further studies. The question could be how sensitive the self-learning MLM is towards the possible anomalous pixels. Could the implementation in *B* be extended for anomaly detection, using the variance and thresholds method with the output distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$, as seen in (Raita-Hakola and Pölönen, 2021).

5. CONCLUSIONS

There is a need for fast, accurate and computationally efficient machine learning methods in hyperspectral imaging remote sensing applications. In this study, we implemented and tested the nearest neighbour Minimal Learning Machine (NN-MLM) with Recursive Least Squares (RLS) updates, and as a result, we created a new method, the self-learning MLM.

The self-learning method was tested in two experiments. Experiment *A* confirms that the updated model can recognise and classify new classes. The classification accuracy increase when the training set reference subset's size increases. The limitation to this approach is the number of new classes. The updated model can handle one class relatively well, but the accuracy decreases after every new class and update. Further research is needed for achieving better overall accuracy with multiple new classes.

The second experiment was more successful than experiment *A*. Experiment *B* simulated the push broom spectral imagers working principles. The model was updated and tested based on a stream of pixel spectrum lines and a continuous scanning process. The results show that it is possible to train the model with a significantly small amount of labelled reference points and update it continuously with (RLS) to reach quickly high classification accuracy. The accuracy results are comparable with NN-MLM, and the benefit is that the new self-learning model can be first trained even with 20 or 100 reference pixel spectra instead of the whole test data.

ACKNOWLEDGEMENTS

This study is partly funded by the Academy of Finland (Grant No. 327862). We thank our colleague Kimmo Riihiaho for proofreading assistance and comments that improved the manuscript.

REFERENCES

Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R. and Sousa, J. J., 2017. Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. Remote Sensing.

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. M. and Chanussot, J., 2013. Hyperspectral remote sensing data analysis and future challenges. IEEE Geoscience and Remote Sensing Magazine 1(2), pp. 6–36.

Caba, J., Díaz, M., Barba, J., Guerra, R., de la Torre, J. A. and López, S., 2020. Fpga-based on-board hyperspectral imaging compression: Benchmarking performance and energy efficiency against gpu implementations. Remote Sensing 12(22), pp. 1–37.

Camps-Valls, G. and Bruzzone, L., 2005. Kernel-based methods for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing 43(6), pp. 1351–1362.

de Souza, A. H., Corona, F., Barreto, G. A., Miche, Y. and Lendasse, A., 2015. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. Neurocomputing 164(January), pp. 34–44.

Graña, M., Veganzons, M. and Ayerdi, B., 2022. Hyperspectral Remote Sensing Scenes. Data available in http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Hakola, A.-M. and Pölönen, I., 2020. Minimal learning machine in hyperspectral imaging classification. In: L. B. Santi, F. Bovolo and Emanuele (eds), Image and Signal Processing for Remote Sensing XXVI, Vol. 11533, p. 26.

Haut, J. M., Paoletti, M. E., Plaza, J. and Plaza, A., 2018. Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines. Journal of Real-Time Image Processing 15(3), pp. 439–462.

Haykin, S. S., 2008. Adaptive filter theory. Pearson Education India.

Honkavaara, E., Saari, H., Kaivosoja, J., Pölönen, I., Hakala, T., Litkey, P., Mäkynen, J. and Pesonen, L., 2013. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. Remote Sensing 5(10), pp. 5006–5039.

Kärkkäinen, T., 2019. Extreme minimal learning machine: Ridge regression with distance-based basis. Neurocomputing 342, pp. 33–48.

Mesquita, D. P., Gomes, J. P. and Souza Junior, A. H., 2017. Ensemble of Efficient Minimal Learning Machines for Classification and Regression. Neural Processing Letters 46(3), pp. 751–766.

Meyer, C. D., 2000. Matrix analysis and applied linear algebra. Vol. 71, Siam.

Nevalainen, O., Honkavaara, E., Tuominen, S., Viljanen, N., Hakala, T., Yu, X., Hyyppä, J., Saari, H., Pölönen, I., Imai, N. N. and Tommaselli, A. M., 2017. Individual tree detection and classification with UAV-Based photogrammetric point clouds and hyperspectral imaging. Remote Sensing.

Nezami, S., Khoramshahi, E., Nevalainen, O., Pölönen, I. and Honkavaara, E., 2020. Tree species classification of drone hyperspectral and RGB imagery with deep learning convolutional neural networks. Remote Sensing.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, pp. 2825–2830.

Pölönen, I., Riihiaho, K., Hakola, A.-M. and Annala, L., 2020. Minimal learning machine in anomaly detection from hyperspectral images. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 43(B3), pp. 467–472.

Raita-Hakola, A.-M. and Pölönen, I., 2021. Piecewise anomaly detection using minimal learning machine for hyperspectral images. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-3-2021, pp. 89–96.

Romberg, J., 2016. Ece 6250 fall 2016 lecture notes. Lecture notes available from Georgia Tech University.

Sawant, S. S. and Manoharan, P., 2020. Unsupervised band selection based on weighted information entropy and 3D discrete cosine transform for hyperspectral image classification. International Journal of Remote Sensing 41(10), pp. 3948–3969.

Tuominen, S., Näsi, R., Honkavaara, E., Balazs, A., Hakala, T., Viljanen, N., Pölönen, I., Saari, H. and Ojanen, H., 2018. Assessment of classifiers and remote sensing features of hyperspectral imagery and stereo-photogrammetric point clouds for recognition of tree species in a forest area of high species diversity. Remote Sensing 10(5), pp. 1–28.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, pp. 261–272.

Xie, F., Li, F., Lei, C., Yang, J. and Zhang, Y., 2019. Unsupervised band selection based on artificial bee colony algorithm for hyperspectral image classification. Applied Soft Computing Journal 75, pp. 428–440.