Extending the Minimal Learning Machine for Pattern Classification

Amauri H. Souza Júnior
Department of Computer Science
Federal Institute of Ceará
Maracanaú, Brazil

Francesco Corona, Yoan Miché and Amaury Lendasse
Department of Information and Computer Science
Aalto University
Espoo, Finland

Guilherme A. Barreto
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil

***Abstract*—The Minimal Learning Machine (MLM) has been recently proposed as a novel supervised learning method for regression problems aiming at reconstructing the mapping between input and output distance matrices. Estimation of the response is then achieved from the geometrical configuration of the output points. Thanks to its comprehensive formulation, the MLM is inherently capable of dealing with nonlinear problems and multidimensional output spaces. In this paper, we introduce an extension of the MLM to classification tasks, thus providing a unified framework for multiresponse regression and classification problems. On the basis of our experiments, the MLM achieves results that are comparable to many de facto standard methods for classification with the advantage of offering a computationally lighter alternative to such approaches.**

I. Introduction

Classification takes an important role in supervised learning. The problem consists in identifying to which of a set of categories (classes) a new observation belongs, on the basis of a training dataset containing observations whose category membership is known. When observations belong to two classes only, the problem is referred to as binary classification; naturally when there are more than two possible classes, it corresponds to a multi-class problem. Among the state-of-the-art methods for classification, we could mention the MultiLayer Perceptron (MLP, [1]); the Support Vector Machine (SVM, [2]), Gaussian Processes (GP, [3]) and the Extreme Learning Machine (ELM, [4]).

The Minimal Learning Machine (MLM, [5]) is a recently proposed method for supervised learning. The basic idea behind the Minimal Learning Machine is the existence of a mapping between the geometric configurations of points in the input and output space. Such a mapping can be reconstructed by learning a multi-response linear regression model between distance matrices. Under these conditions, for an input point with known configuration in the input space, its corresponding configuration in the output space can be easily estimated after learning a simple linear model between input and output distance matrices. The resulting estimate is then used to locate the output point and thus provide an estimate for the response.

Even though the Minimal Learning Machine was proposed to deal with both regression and classification, the MLM has not been thouroughly evaluated on classification tasks. In this work, we discuss a natural extension of the MLM to multi-class classification and we evaluate its performance on real-world classification problems. On the basis of our experiments, the MLM achieves accuracies comparable and even better than those obtained with state-of-the-art methods and it still offers a computationally light alternative to such approaches.

The remainder of the paper is organized as follows. Section II overviews the Minimal Learning Machine; the general formulation of the MLM is discussed (Section II-A) and its properties presented (Section II-B). In Section III, we propose an extension of the MLM to classification problems along with an illustrative example (Section III-A). In Section IV, a thorough experimental assessment of the Minimal Learning Machine is conducted to evaluate its performance and to compare it with state-of-the-art approaches in classification tasks.

II. Minimal Learning Machine

In this section, we formulate the Minimal Learning Machine (MLM, [5]) and we discuss its computational complexity and its (hyper-)parameters.

A. Formulation

We are given a set of N input points $X = \{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^D$, and the set of corresponding outputs $Y = \{\mathbf{y}_i\}_{i=1}^N$, with $\mathbf{y}_i \in \mathbb{R}^S$. Assuming the existence of a continuous mapping $f : \mathcal{X} \to \mathcal{Y}$ between the input and the output space, we want to estimate f from data with the multiresponse model

$$\mathbf{Y} = f(\mathbf{X}) + \mathbf{R}.$$

The columns of the matrix $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(D)}]$ correspond to the D inputs and the rows to the N observations. Equally, the columns of the matrix $\mathbf{Y} = [\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(S)}]$ correspond to the S outputs and the rows to the N observations. The columns of the $N \times S$ matrix $\mathbf{R}$ correspond to the residuals.

The MLM is a two-step method designed to

1) reconstruct the mapping existing between input and output distances;

978-1-4799-3194-1/13 $31.00 © 2013 IEEE
DOI 10.1109/BRICS-CCI-CBIC.2013.46

IEEE computer society

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.

2) estimating the response from the configuration of the output points.

In the following, the two steps are discussed.

1) Distance regression: For a selection of reference input points $R = \{\mathbf{m}_k\}_{k=1}^{K}$ with $R \subseteq X$ and corresponding outputs $T = \{\mathbf{t}_k\}_{k=1}^{K}$ with $T \subseteq Y$, define $\mathbf{D}_x \in \mathbb{R}^{N\times K}$ in such a way that its kth column contains the distances $d(\mathbf{x}_i, \mathbf{m}_k)$ between the N input points $\mathbf{x}_i$ and the kth reference point $\mathbf{m}_k$. Analogously, define $\mathbf{\Delta}_y \in \mathbb{R}^{N\times K}$ in such a way that its kth column contains the distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$ between the N output points $\mathbf{y}_i$ and the output $\mathbf{t}_k$ of the kth reference point. The mapping g between the input distance matrix $\mathbf{D}_x$ and the corresponding output distance matrix $\mathbf{\Delta}_y$ can be reconstructed using the multiresponse regression model

$$\mathbf{\Delta}_y = g(\mathbf{D}_x) + \mathbf{E}.$$

The columns of the matrix $\mathbf{D}_x = [d(\mathbf{x}, \mathbf{m}_1), \ldots, d(\mathbf{x}, \mathbf{m}_K)]$ correspond to the K input vectors and the columns of the matrix $\mathbf{\Delta}_y = [\delta(\mathbf{y}, \mathbf{t}_1), \ldots, \delta(\mathbf{y}, \mathbf{t}_K)]$ correspond to the K response vectors, the N rows correspond to the observations. The columns of the $N \times K$ matrix $\mathbf{E}$ correspond to the K residuals.

Assuming that mapping g between input and output distance matrices has a linear structure for each response, the regression model has the form

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

The columns of the $K \times K$ regression matrix $\mathbf{B}$ correspond to the coefficients for the K responses. The matrix $\mathbf{B}$ can be solved from data through a minimization of the multivariate residual sum of squares as loss function:

$$\begin{aligned} \mathrm{RSS}(\mathbf{B}) &= \sum_{k=1}^{K}\sum_{i=1}^{N}\Big(\delta(\mathbf{y}_i, \mathbf{t}_k) - g_k(d(\mathbf{x}_i, \mathbf{m}_k))\Big)^2 && \text{(2a)} \\ &= \mathrm{tr}\Big((\mathbf{\Delta}_y - \mathbf{D}_x\mathbf{B})'(\mathbf{\Delta}_y - \mathbf{D}_x\mathbf{B})\Big) && \text{(2b)} \end{aligned}$$

Under the normal conditions where the number of equations in Equation 1 is larger to the number of unknowns, the problem is overdetermined and, usually, with no solution. This corresponds to the case where the number of selected reference points is smaller than the number of available points available (i.e., $K < N$). In this case, we have to rely on the approximate solution provided by the usual least squares estimate of $\mathbf{B}$,

$$\hat{\mathbf{B}} = (\mathbf{D}_x'\mathbf{D}_x)^{-1}\mathbf{D}_x'\mathbf{\Delta}_y. \tag{3}$$

If in Equation 1 the number of equations equals the number of unknowns (i.e., $K = N$ because all the learning points are also reference points), then the problem is uniquely determined and, usually, with a single solution:

$$\hat{\mathbf{B}} = (\mathbf{D}_x)^{-1}\mathbf{\Delta}_y. \tag{4}$$

Clearly less interesting is the case where in Equation 1 the number of equations is smaller than the number of unknowns (i.e., for $K > N$, corresponding to the situation where, after selecting the reference points, only a smaller number of learning points is used). This case leads to an underdetermined problem with, usually, infinitely many solutions.

Given the possibility for $\mathbf{B}$ to be either uniquely solvable (Equation 4) or be estimated (Equation 3), for an input test point $\mathbf{x} \in \mathbb{R}^D$ whose distances from the K reference input points $\{\mathbf{m}_k\}_{k=1}^{K}$ are collected in the vector $\mathbf{d}(\mathbf{x}, R) = [d(\mathbf{x}, \mathbf{m}_1) \ldots d(\mathbf{x}, \mathbf{m}_K)]$, the corresponding distances between its unknown output $\mathbf{y}$ and the known outputs $\{\mathbf{t}_k\}_{k=1}^{K}$ of the reference points is

$$\hat{\boldsymbol{\delta}}(\mathbf{y}, T) = \mathbf{d}(\mathbf{x}, R)\hat{\mathbf{B}}. \tag{5}$$

The vector $\hat{\boldsymbol{\delta}}(\mathbf{y}, T) = [\hat{\delta}(\mathbf{y}, \mathbf{t}_1) \ldots \hat{\delta}(\mathbf{y}, \mathbf{t}_K)]$ provides an estimate of the geometrical configuration of $\mathbf{y}$ and the reference set T, in the $\mathcal{Y}$-space.

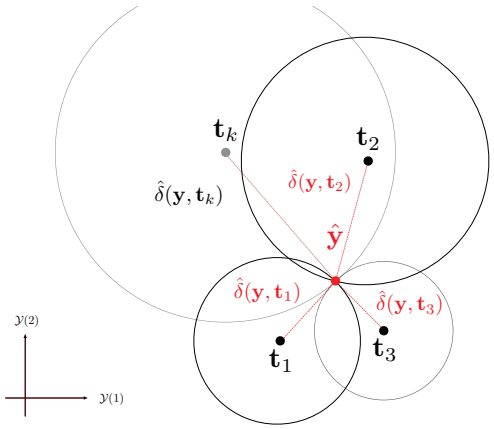


Fig. 1. Output estimation.

2) Output estimation: The problem of estimating the output $\mathbf{y}$, given the outputs $\{\mathbf{t}_k\}_{k=1}^{K}$ of all the reference points and an estimate $\hat{\boldsymbol{\delta}}(\mathbf{y}, T)$ of their mutual distances, can be understood as a multilateration [6] problem to estimate its location in $\mathcal{Y}$. The problem of locating $\mathbf{y}$ is equivalent to solve the overdetermined set of nonlinear equations corresponding to $(S+1)$-dimensional hyperspheres centered in $\mathbf{t}_k$ and passing through $\mathbf{y}$, that is with a radius equal to $\hat{\delta}(\mathbf{y}, \mathbf{t}_k)$:

$$(\mathbf{y} - \mathbf{t}_k)'(\mathbf{y} - \mathbf{t}_k) = \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k), \quad \forall k = 1, \ldots, K. \tag{6}$$

The problem in (6) can be formulated as an optimization problem, then an estimate $\hat{\mathbf{y}}$ can be obtained by the following minimization:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\mathrm{argmin}} \sum_{k=1}^{K}\Big((\mathbf{y} - \mathbf{t}_k)'(\mathbf{y} - \mathbf{t}_k) - \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k)\Big)^2. \tag{7}$$

The objective has a minimum equal to 0 that can be achieved if and only if $\mathbf{y}$ is the solution of (6). If it exists, such a solution is global and unique. Due to the uncertainty introduced by the estimates $\hat{\delta}(\mathbf{y}, \mathbf{t}_k)$, an optimal solution to (7) can still be achieved using standard gradient descent methods. In the following, Levenberg-Marquardt (LM) method [7] is used throughout the experiments.

The MLM method for training and testing are sketched in Algorithm 1 and 2, respectively.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 MLM training procedure

Input: Training data sets X and Y, and K.
Output: $\hat{\mathbf{B}}$, R and T.

1. Randomly select K reference points, R, from X and their corresponding outputs, T, from Y;
2. Compute $\mathbf{D}_x$: The distance matrix between X and R;
3. Compute $\mathbf{\Delta}_y$: The distance matrix between Y and T;
4. Calculate $\hat{\mathbf{B}} = (\mathbf{D}_x'\mathbf{D}_x)^{-1}\mathbf{D}_x'\mathbf{\Delta}_y$.

Algorithm 2 MLM test procedure

Input: $\hat{\mathbf{B}}$, R, T and $\mathbf{x}$.
Output: $\hat{\mathbf{y}}$.

1. Compute $\mathbf{d}(\mathbf{x}, R)$;
2. Compute $\hat{\boldsymbol{\delta}}(\mathbf{y}, T) = \mathbf{d}(\mathbf{x}, R)\hat{\mathbf{B}}$;
3. Use T and $\hat{\boldsymbol{\delta}}(\mathbf{y}, T)$ to find an estimate for $\mathbf{y}$. This can be accomplished by any gradient descent algorithm over the cost function in Eq. 7.

B. Parameters and computational complexity

Hyper-parameters: On the basis of the aforementioned overview, the number of reference points K is virtually the only hyper-parameter that the user needs to select in order to optimize a Minimal Learning Machine. As always, a selection based on conventional validation or on standard resampling methods for cross-validation could be adopted for the task.

Two figures of merit are considered for selecting K, the Average Mean Squared Error for the output distances ($AMSE(\boldsymbol{\delta})$) and the Average Mean Squared Error for the responses ($AMSE(\mathbf{y})$):

$$AMSE(\boldsymbol{\delta}) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_v}\sum_{i=1}^{N_v}(\delta(\mathbf{y}_i, \mathbf{t}_k) - \hat{\delta}(\mathbf{y}_i, \mathbf{t}_k))^2 \quad (8)$$

$$AMSE(\mathbf{y}) = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{N_v}\sum_{i=1}^{N_t}(y_i^{(s)} - \hat{y}_i^{(s)})^2 \quad (9)$$

For a set of N_v validation points $(\mathbf{x}_i, \mathbf{y}_i)$, the $AMSE(\boldsymbol{\delta})$ quantifies how well the distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$ between the N_v output responses $\mathbf{y}_i$ and the outputs of the K selected reference points $\mathbf{t}_k$ are estimated $\hat{\delta}(\mathbf{y}_i, \mathbf{t}_k)$, after the distance regression step of the MLM and before the estimation is even performed. The $AMSE(\mathbf{y})$, on the other hand is only performed after both the distance regression and the estimation steps of the MLM are completed and thus it quantifies how well the S-dimensional outputs $y_i^{(s)}$ are estimated $\hat{y}_i^{(s)}$. In the case of univariate responses ($S = 1$) the $AMSE(\mathbf{y})$ reduces to the conventional Mean Square Error for the outputs ($MSE(y)$).

Computational Complexity: The Minimal Learning Machine training computation can be roughly decomposed into two steps: i) calculations of the pairwise distance matrices in the output and input space; ii) calculation of the least-square solution for the multiresponse linear regression problem on distance matrices (see Eq. 3).

The first procedure takes $\Theta(KN)$ time, see [8] for a review of algorithmic asymptotic analysis. The computational cost of the second step is driven by the calculation of the Moore-Penrose pseudoinverse matrix. One of the most used method for the task is the SVD [9], which runs in $\Theta(K^2N)$ time. This method is very accurate but its drawback relies in the computational time constants that makes the method time-intensive. Several methods have been proposed in order to speed up the computation of the Moore-Penrose matrix (for example, see [10], [11]). In [10], the computation is optimized by using a special type of tensor product and QR factorization whereas the method proposed in [11] is based on a full-rank Cholesky decomposition. In spite of such approaches improve significantly the computational time of computing the Moore-Penrose inverse matrix, the time complexity is still equal to that provided by the SVD method. Even though, one might consider them for large datasets and real-time applications.

The time complexity of the MLM training phase is driven by the computation of the Moore-Penrose matrix and then it is given by $\Theta(K^2N)$. In order to establish a comparison, the MLM training computational cost is similar to what presented by an Extreme Learning Machine when the number of hidden neurons is equal to the number of reference points. It is worthy to notice that the ELM is considered one of the fastest methods for nonlinear regression and classification tasks [12].

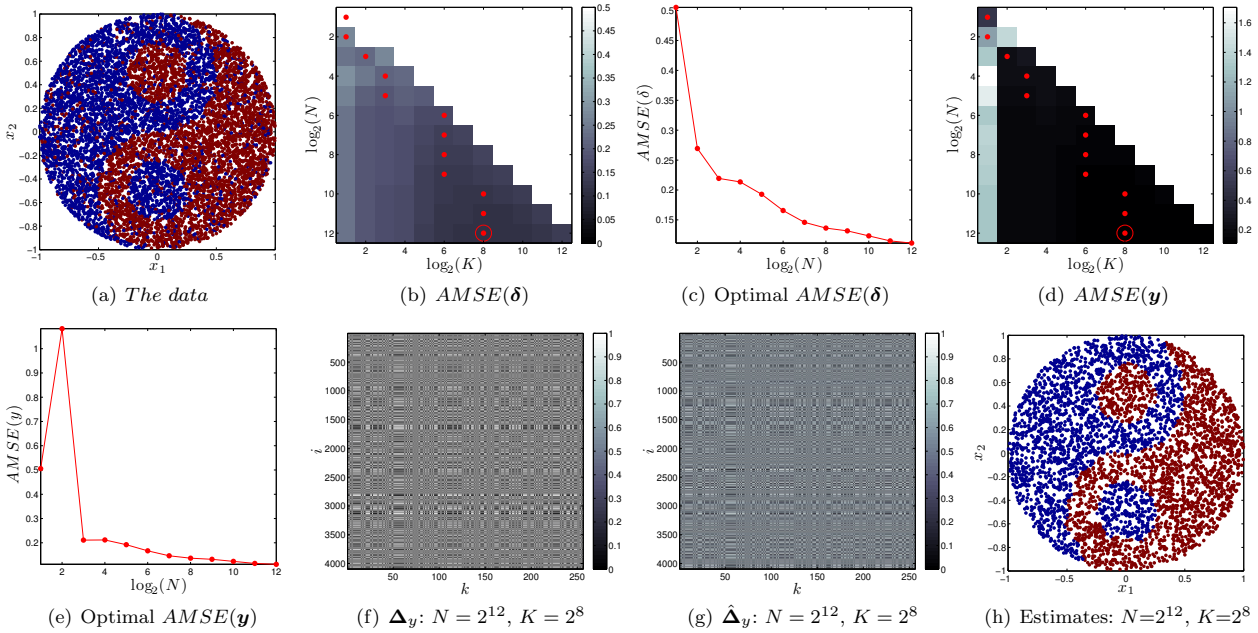
Concerning the computational analysis of the generalization step in a MLM, we consider the Levenberg-Marquardt method due to its fast and stable convergence, even though any gradient descent method can be used on the minimization step in Eq. 7. For each iteration, the LM method involves the computation of the Jacobian matrix and its inverse. In this regard, the computational complexity of the LM algorithm is about $\Theta(S^3)$, where S is the dimensionality of $\mathbf{y}$. In most of the regression and classification problems, S is a small number and then the cost function evaluation (Eq. 7) is the most computationally demanding operation and it is proportional to the number of reference points.

III. The Minimal Learning Machine for classification

An important class of problems is classification, where we are concerned about predicting categories usually denoted by qualitative outputs, also called class labels. For the task, we are still given a set of N input points $X = \{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^D$, and the set of their corresponding class labels $L = \{l_i\}_{i=1}^N$, with $l_i \in \{C_1, \ldots, C_S\}$, where C_j denotes the j-the class; for $S = 2$, the problem is referred to as binary classification, whereas for $S > 2$ we have multiclass applications.

The Minimal Learning Machine can be extended to classification problems in a straightforward manner by representing the S class labels in a vectorial fashion through an 1-of-S encoding scheme. In such approach, a S-level qualitative variable is represented by a vector of S binary variables or bits, only one of which is *on* at a time. Mathematically, the set of outputs $Y = \{\mathbf{y}_i\}_{i=1}^N$, with $\mathbf{y}_i \in \mathbb{R}^S$, that corresponds to the input points X is then defined in such a way that the jth-component of $\mathbf{y}_i$ is set to α if $l_i = C_j$ and β otherwise, where α and β are

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.



(a) *The data* (b) $AMSE(\boldsymbol{\delta})$ (c) Optimal $AMSE(\boldsymbol{\delta})$ (d) $AMSE(\boldsymbol{y})$

(e) Optimal $AMSE(\boldsymbol{y})$ (f) $\boldsymbol{\Delta}_y$: $N = 2^{12}$, $K = 2^8$ (g) $\hat{\boldsymbol{\Delta}}_y$: $N = 2^{12}$, $K = 2^8$ (h) Estimates: N=2^{12}, K=2^8

Fig. 2. The Tai Chi example.

integer scalars such as $\alpha > \beta$. An usual choice is $\alpha = 1$ and $\beta = -1$.

In classification of a test observation $\mathbf{x}$ with unknown class label $l \in \{C_1, \ldots, C_S\}$, the estimated class $\hat{l}$ associated to the output estimate $\hat{\mathbf{y}}$ is given by $\hat{l} = C_{s^*}$, where

$$s^* = \underset{s=1,\ldots,S.}{\operatorname{argmax}} \{\hat{y}^{(s)}\}. \tag{10}$$

As one can easily notice, for binary classification problems, we may simplify the approach by using a binary single output scheme where the outputs are represented by scalars $y_i \in \{\alpha, \beta\}$ in correspondence to the two classes.

Given this formulation, the Minimal Learning Machine provides unified implementation for regression, binary and multi-class applications.

A. An illustrative example

In this section, we illustrate the Minimal Learning Machine and its properties on a classification example, the Tai Chi symbol. For the task, we generated 2^{13} bidimensional input points uniformly distributed in the Tai Chi symbol, and after assigning the class labels to the Yin an Yang areas we purposely mislabeled 10% of the observations, Figure 2(a). Half of the whole dataset is used for learning and the 2^{12} remaining samples are used for validation purposes. The performance of the MLM is analyzed for a number N of learning points and a number K of randomly selected reference points in $\{2^1, 2^2, \ldots, 2^{12}\}$. Moreover, for each size N of the learning set, we always selected a number K of reference points such that $K \leq N$.

We evaluate the MLM on the validation set using the two figures of merit for selecting K; the $AMSE(\boldsymbol{\delta})$ and $AMSE(\mathbf{y})$. In our experiments, Minimal Learning Machines are trained with all the different N-sized learning sets and for all possible number of reference points. That is, for each size N of the learning set, the accuracies of the distance regression and the output estimation steps are depicted for a varying number K of reference points.

Figure 2(b) shows the results regarding the $AMSE(\boldsymbol{\delta})$, where for a given number of learning points, the model achieving the best performance is denoted by a red dot. A red circle is used to denote the model with the overall smallest $AMSE(\boldsymbol{\delta})$. Figure 2(c) show the accuracies of the best performing models per size of the learning set. One might observe that such accuracies improve as the number of learning points increases, that is not the case for the number of reference points since the best model overall has $K = 2^8$. Figure 2(d) shows the results with respect to the $AMSE(\mathbf{y})$, where again the best performing models are denoted by red dots, and a red circle denotes the best model overall. Equally, Figure 2(e) show the accuracies of the best models for each learning set.

It is worth noticing that in both distance regression and output estimation steps, the optimal combination for N and K is the same: $N = 2^{12}$ and $K = 2^8$. Given such behavior we suggest to select K based on the distance regression step and thus avoiding the need to also perform the optimization.

To assign a visual evaluation on the regression step, we report in Figures 2(f) and 2(g) the actual and estimated pairwise output distance matrices respectively, where it is possible to notice that such distances are correctly reconstructed and the overall structure of $\boldsymbol{\Delta}_y$ is preserved in $\hat{\boldsymbol{\Delta}}_y$. With respect to the estimation step, Figure 2(h) shows the estimated classes in validation using the best model; the

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.

TABLE I. TEST PERFORMANCE: ACCURACIES (%), THE CORRESPONDING STANDARD DEVIATIONS AND t-TEST RESULTS (✓ FOR ACCEPT, × FOR REJECT AND p-VALUES). FOR EACH DATASET, THE BEST PERFORMING MODELS ARE IN BOLDFACE.

Datasets	**Models**					
	MLM	ELM	OPELM	SVM	GP	MLP
Wisconsin B. C.	**97.7** **0.6**	95.6 1.2 × (3e-4)	91.6 1.7 ×(2e-4)	91.6 1.7 ×(2e-9)	97.3 0.9 ✓(.30)	96.6 1.9 ×(4e-3)
Pima I. D.	74.2 1.7	72.2 1.9 ×(.02)	74.9 2.4 ✓(.45)	72.7 1.5 ✓(.05)	**76.3** **1.8** ×(.02)	75.2 1.9 ✓(.25)
Iris	95.0 1.4	72.2 1.0 ×(1e-6)	95.0 2.1 ✓(1)	95.4 1.9 ✓(.60)	**95.6** **2.3** ✓(.49)	94.8 3.8 ✓(.88)
Wine	**99.0** **1.2**	81.8 6.2 ×(8e-8)	90.7 4.9 ×(5e-5)	95.8 2.9 ×(4e-3)	96.2 2.1 ×(e-3)	96.0 2.4 ×(2e-3)

accuracy is 88%. Interestingly, the error tends to the the percentage of mislabeled data, then corresponding to an effective model that does not suffer from overfitting.

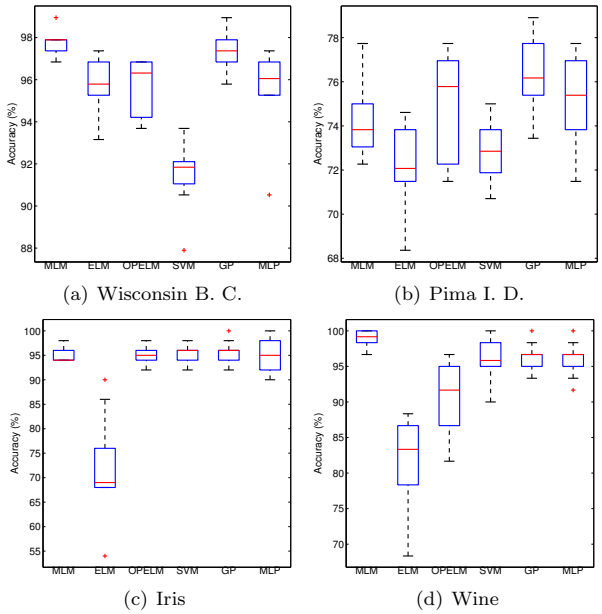
IV. EXPERIMENTS

In this section, we present the results achieved by the Minimal Learning machine on four real-world datasets commonly used for benchmarking purposes in classification. The performance of the MLM is then compared to what achieved with five other reference methods: The Extreme Learning Machine (ELM, [4]), the Optimally Pruned ELM (OP-ELM [12]), the Support Vector Machine (SVM, [2]), Gaussian Processes (GP, [3]) and the MultiLayer Perceptron (MLP, [1]).

The datasets are available from the University of California at Irvine (UCI) Repository (`www.ics.uci.edu/~mlearn/`). The datasets used in the experiments consist of both binary and multi-class problems and they are: 1) Wisconsin Breast Cancer (30 inputs, 2 classes, 569 samples); 2) Pima Indians Diabetes (8 inputs, 2 classes, 768 samples); 3) Iris (4 inputs, 3 classes, 150 samples) and 4) Wine (13 inputs, 3 classes, 178 samples). All the datasets have been preprocessed in the same way. Ten different random permutations of the whole dataset are taken, and two thirds are used to create the training set and the remaining for the test set. Then, the training set is normalized to zero mean and unit variance, and the test set is normalized using the same mean and variance from the training set. The proportions of the classes are kept balanced: each class is represented in an equal proportion, in both training and test sets.

The hyper-parameters for the SVM and the MLP are selected using 10-fold cross-validation. The SVM is learned using the SVM toolbox [13] with default settings for the hyper-parameters and grid search: the grid is logarithmic between 2^{-2} and 2^{10} for each hyper-parameter and radial basis function kernel. The MLP is trained using the Levenberg-Marquardt optimization and a range of hidden units from 1 to 20. The learning of GP is based on the default settings in the Matlab Toolbox [3]. The ELM and OP-ELM have been validated using sigmoid, gaussian and linear kernels, and a maximum number of 100 hidden units. As for the Minimal Learning Machine, the only hyper-parameter of the method (the number K of reference points) has also been selected through 10-fold cross-validation, for a K ranging from 5% to 100% (with a step size of 5%) of the available training samples.

In order to evaluate the MLM performance for classification problems, the mean success classification rate and the corresponding standard deviations for ten different dataset permutations. In addition, objecting to compare the accuracy achieved by the MLM to all the other methods from a statistical point of view, we carried out the two-sample t-test. The null hypothesis corresponds to independent gaussian random variables with equal means and equal but unknown variances. The results are reported in Table I.



(a) Wisconsin B. C. (b) Pima I. D.

(c) Iris (d) Wine

Fig. 3. Box-plots of models accuracies.

From Table I we can observe that the MLM exhibits an equivalent or even better generalization performance in comparison to the other models. Moreover, the MLM has shown a stable performance since its standard deviations are smaller than that of the other methods specially on the Wisconsin B. C. and Wine datasets. The accuracies

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.

for all the methods and datasets are shown through boxplots in Figure 3. Based on the hypothesis tests, the GP again achieved results quite similar to those from MLM and it is represented by the equivalence for three datasets: Wisconsin B. C., Pima I. D. and Iris. Concerning the Wine dataset, the MLM is the best performing model, since it has smallest accuracy value and the null hypotheses were rejected for all the models.

As one can notice, the MLM performance is quite similar to those state-of-the-art methods. Then, the computational complexity takes an important role in the decision making process of selecting the most appropriate method. In this regard, an essential aspect for a fast MLM training is the number of reference points, or more specifically, we are interested in the property that the optimal number of reference points does not grow at the same rate of the number of learning points (dataset size). Thus, to illustrate such property we report in Figure 4 the normalized MSE (NMSE) per number of reference points in the cross-validation process.

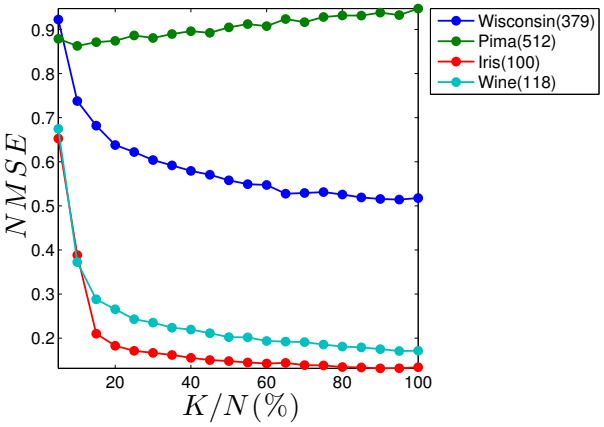


Fig. 4. Validation results: error per number of reference points. Legends also contain the total number of training samples.

From Figure 4, our experiments have shown that it is not needed as many reference points as learning points, specially for the Pima Indians Diabetes data. For all datasets, 20% of the number of learning points has provided a good threshold for selecting K.

V. Conclusions

This work overviews a new supervised learning method, the Minimal Learning Machine, MLM. Learning a MLM consists in reconstructing the mapping existing between input and output distance matrices and then exploiting the geometrical arrangement of the output points for estimating the response. Based on our experiments, a multiresponse linear regression model is capable to reconstruct the mapping existing between the aforementioned distance matrices. The MLM has only one hyper-parameter to be optimized using standard resampling methods, like LOO cross-validation. Given its general formulation, the Minimal Learning Machine is also inherently capable to operate on multidimensional responses and it can be extended to classification problems in a straightforward fashion.

On a large number of synthetic and real-world problems, the Minimal Learning Machine has achieved accuracies that are comparable to what is obtained using state-of-the art classification methods. For compactness, we have reported the performances on a selection of four datasets from the UCI Repository and comparisons with five reference approaches. The results highlight the potentiality of the MLM on classification tasks.

Acknowledgment

The authors would like to thank the financial support received from the Brazilian Agency of Post-Graduate Studies (CAPES) under the grant number 9147-12-8.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY: Oxford University Press, Inc., 1995.

[2] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press, 2006.

[4] G. B. Huang, Q. Y. Zhu, and C. K. Ziew, "Extreme Learning Machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[5] A. H. Souza Junior, F. Corona, Y. Miché, A. Lendasse, G. Barreto, and O. Simula, "Minimal learning machine: A new distance-based method for supervised learning," in *Proceedings of the 12th International Work Conference on Artificial Neural Networks (IWANN'2013)*, ser. Lecture Notes in Computer Science, vol. 7902. Springer, 2013, pp. 408–416.

[6] E. Niewiadomska-Szynkiewicz and M. Marks, "Optimization schemes for wireless sensor network localization," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 291–302, 2009.

[7] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.

[10] V. N. Katsikis, D. Pappas, and A. Petralias, "An improved method for the computation of the moore-penrose inverse matrix," *Applied Mathematics and Computation*, vol. 217, pp. 9828–9834, 2011.

[11] P. Courrieu, "Fast computation of moore-penrose inverse matrices," *Neural Information Processing Letters and Reviews*, vol. 8, pp. 25–29, 2005.

[12] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally Pruned Extreme Learning Machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1–27, 2011.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 21,2022 at 17:28:19 UTC from IEEE Xplore. Restrictions apply.