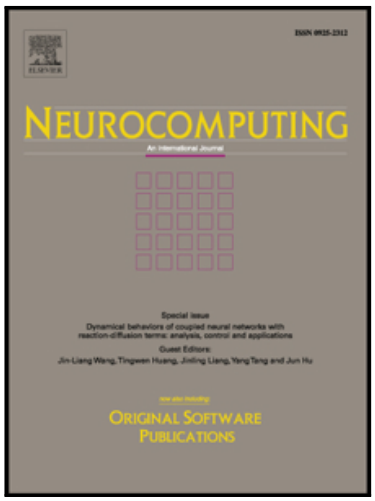

Extreme minimal learning machine: Ridge regression with distance-based basis

Tommi Kärkkäinen

PII: S0925-2312(19)30143-2
DOI: https://doi.org/10.1016/j.neucom.2018.12.078
Reference: NEUCOM 20414

To appear in: *Neurocomputing*

Received date: 7 July 2018
Revised date: 21 December 2018
Accepted date: 23 December 2018

Please cite this article as: Tommi Kärkkäinen, Extreme minimal learning machine: Ridge regression with distance-based basis, *Neurocomputing* (2019), doi: https://doi.org/10.1016/j.neucom.2018.12.078

Extreme minimal learning machine: Ridge regression with distance-based basis

Tommi Kärkkäinen[a]

[a] *University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyvaskyla, Finland*

Abstract

The extreme learning machine (ELM) and the minimal learning machine (MLM) are nonlinear and scalable machine learning techniques with a randomly generated basis. Both techniques start with a step in which a matrix of weights for the linear combination of the basis is recovered. In the MLM, the feature mapping in this step corresponds to distance calculations between the training data and a set of reference points, whereas in the ELM, a transformation using a radial or sigmoidal activation function is commonly used. Computation of the model output, for prediction or classification purposes, is straightforward with the ELM after the first step. In the original MLM, one needs to solve an additional multilateration problem for the estimation of the distance-regression based output. A natural combination of these two techniques is proposed and experimented here: to use the distance-based basis characteristic in the MLM in the learning framework of the regularized ELM. In other words, we conduct ridge regression using a distance-based basis. The experimental results characterize the basic features of the proposed technique and surprisingly, indicate that overlearning with the distance-based basis is in practice avoided in classification problems. This makes the model selection for the proposed method trivial, at the expense of computational costs.

Keywords: Randomized learning machines, Extreme learning machine, Minimal learning machine, Extreme minimal learning machine

1. Introduction

Kernels and basis functions have a central role in machine learning. The appearance of the radial basis function network (RBFN) in the 1980s [1–3] made it clear that universal approximation property of a neural network technique does not need a fully adaptable basis. With an a priori fixed location and the scatter parameters of radial basis functions, one could construct nonlinear approximators of unknown functions for regression and classification. In the work of Kwok and Yeung [4] something similar was suggested for the multilayered perceptron, MLP: First optimize all weights using the whole data and then freeze the hidden layer weights in the nonlinear cross-validation, by adapting only the weights in the outer layer. The approach to sequentially separate the learning of weights in the outer and hidden layer for a single-hidden-layer-feedforward-network (SLFN) was proposed and tested by McLoone et al. [5] (see also [6]).

In the MLP and in deep learning [7, 8], we might have a large pool of adaptation in the deeply layered basis. However, the extreme learning machine, ELM, as proposed by Huang et al. [9, 10], established the key randomized neural network framework without kernel adaptation [11]. Actually, the first step of the expectation-maximization (EM) approach proposed in McLoone et al. [5] coincides with the basic definition of the ELM. As explained and thoroughly described by Cao et al. [12], this training mechanism can also be traced back to random vector functional link (RVFL) networks [13] and Schmidt's method [14]. The ELM provides a simple, but still universal, approach to nonlinear, data-based modeling through generation of the hidden layer weights [15]. More recently, the universal approximation properties of the ELM were revisited by presenting probabilistic convergence analysis [16, 17]. There, the necessity of the repeated sampling of the sigmoidal kernel and the advantage of the weight decay (ridge regression) were concluded. ELM techniques have been used extensively and successfully in different fields of applications [18–20].

Another novel supervised learning method with a random basis, the minimal learning machine (MLM), was proposed in the works of de Souza Junior et al. [21, 22]. The MLM is based on the idea of the existence of mapping between the geometric configurations of input-output points. The original derivation owed much to the classical unsupervised technique of multidimensional scaling (MDS) [23, 24]. The nonlinear geometric configuration in the MLM is learned using a distance-based regression technique, where reference point subsets are first sampled from input and output data. Then, two distance/dissimilarity matrices are formed between the reference points and the training data. For the output of any test observation, the computed distances are used to define a multilateration problem, of the same form as the squared stress formu-

Email address: tommi.karkkainen@jyu.fi (Tommi Kärkkäinen)

lation for the MDS [25], that needs to be solved. Possibilities for supervised learning with missing input/feature values [26, 27] increased interest in MLM. As described by de Souza Junior et al. [22], the dissimilarities in the MLM are usually computed using the Euclidean distance, but nothing prevents using any dissimilarity measure in a metric space for different types of data. Links between the theory of learning with similarity functions and the ELM were addressed in the work of Gastaldo et al. [28]. The MLM has been recently applied, for example, in human activity recognition [29] and mobile robot localization [30, 31].

Even if the distribution in the ELM, where the hidden layer weights are generated, and the dissimilarity measure in the MLM define a family of methodological variants, in principle, both the ELM and the MLM contain only one hyperparameter: the size of the hidden layer in the ELM or the number of reference points in the MLM. Typical choices of this parameter are proportional to the number of observations available in the training set [10, 22, 32, 33].

In this paper, a natural combination of the ELM and the MLM referred to as the *extreme minimal learning achine (EMLM)*, is proposed and described. The technique uses a distance-based feature mapping originating from the MLM to generate a random basis for a nonlinear approximation of the input data. Then, similarly to the ELM (and many other basically linear techniques [34]), the regularized least-squares problem as in the ridge regression is solved to recover the matrix of weights to combine the distance-based random basis. Compared to the MLM, the solution of the multilateration problem to estimate the actual distance-regression-based output is omitted. Moreover, in the experimental comparison of different methods in the context of classification, we use the fast nearest neighbor (1NN) MLM as derived in [35, Section 3.1].

The rest of the paper is organized as follows. We present the basic formulation of the EMLM based on a unified treatment of the ELM and the MLM in Section 2. Section 3 presents the results and immediate conclusions from the experimental comparison of the ELM, the MLM, and the EMLM. General conclusions and future work are outlined in Section 4.

2. Derivation of the extreme minimal learning machine

In this section, we first introduce the general learning framework, the regularized least-squares optimization formulation, for the extreme learning machine. We then use this formulation to present the MLM and to derive the EMLM as a straightforward combination of the two original methods. Because the MLM was recently thoroughly tested for regression in [36], we restrict ourselves to classification problems.

For this purpose, we let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^k$, be the given training data of the input-output samples. Here, N refers to the number of observations, n denotes the input dimension, i.e., the number of variables, and k the output dimension, i.e., the number of classes. For all the formulations here, the class encoding is realized with the *1-of-k* coding scheme using the standard basis in $\mathbb{R}^k$. Let $\mathbf{X} = \left[\mathbf{x}_i\right]_{i=1}^N \in \mathbb{R}^{n\times N}$ and $\mathbf{Y} = \left[\mathbf{y}_i\right]_{i=1}^N \in \mathbb{R}^{k\times N}$ be the matrix representations of the vector-valued inputs and output encodings sequentially.

2.1. The ELM

In the ELM, the nonlinear random basis can be constructed using many kinds of feature mappings [32, 37, 38]: sigmoid nodes, radial basis nodes, threshold nodes, trigonometric nodes, high-order polynomial nodes, wavelet and Fourier series functions, etc. As summarized by Cao et al. [12], in the case of an especially shallow feedforward network with an appropriate choice (e.g., linear) of feature mapping, many core techniques in machine learning, including support vector machines (SVMs), principal component analysis (PCA), and random projection (RP) (see [19, 39]), can be presented in the general learning framework of the ELM.

Theoretically, the most important tenet of the ELM is the interpolation and universal approximation capability, as analyzed for the radial basis feature mappings in [1–3] and derived and depicted for the ELM in [15–17, 19, 32]. More precisely, in [15] it was shown that a linear combination of either additive nodes of the form $g(\mathbf{a}_i^T\mathbf{x} + b_i)$ or radial basis functions $g\left(\frac{\|\mathbf{x}-\mathbf{a}_i\|}{b_i}\right)$, with randomly generated hidden-layer weights $\{\mathbf{a}_i\}$ and biases $\{b_i\}$ from continuous sampling distributions, can approximate any continuous target function. This holds true when g is a bounded, nonconstant, piecewise continuous activation function for the additive nodes or any integrable, piecewise continuous activation function (with $\int_{\mathbb{R}} g(x)dx \neq 0$) for the radial basis nodes. In this original work, the ranges of the generated weights and biases were not restricted explicitly. In [40] for the sigmoidal activation and more recently in [41] for multiple activation functions, control of the magnitude of weights and biases, for the generated basis functions to act in the nonsaturated region of the feature space, was emphasized. For this work, an interesting activation function satisfying the assumptions of universal approximation [15] is given by the *multiquadric function* $g = (\|\mathbf{x} - \mathbf{a}_i\|^2 + b^2)^{\frac{1}{2}}$ [19, 28, 32].

Another universal approximation analysis track with random basis was given in [16, 17]. In essence, the importance and consequences of the probabilistic facet of the ELM (not visible *per se* when the limit behavior was analyzed in previous papers as summarized above), random generation, were now emphasized [42]. Actually, in [43], a proof of nonconvergence for the classical incremental strategy was given. The analyses in [16, 17], establishing the average convergence bounds in probability, implied that

a) sampling in random generation is necessary because one realization does not guarantee convergence due to the uncertainty problem,

b) cross-validation should be employed to determine the number of random basis functions,

c) regularization (weight decay) techniques provide a remedy for the generalization degradation phenomenon especially with the RBF nodes, and

d) the established generalization capability is valid only for algebraic polynomials, Nadaraya-Watson functions, or sigmoidal activation functions.

The universal consistency analysis and simulated experiments also supported the viability of the sigmoidal activation function [44].

Based on the results as reviewed above, we use the sigmoidal activation function for the ELM. For this purpose, let us attach for each bias-enlarged input vector $\mathbf{x}_i^e = [1\ \mathbf{x}_i^T]^T \in \mathbb{R}^{n+1}$ the sigmoidal basis function

$$\mathbf{h}_i = \frac{1}{1+\exp(-\mathbf{G}\mathbf{x}_i^e)} \in \mathbb{R}^m, \tag{1}$$

where $\mathbf{G} \in \mathbb{R}^{m\times(n+1)}$ with $(\mathbf{G})_{i1} \in \mathcal{U}([0,1])$, $i = 1,\ldots,m$ (the bias weights) and $(\mathbf{G})_{ij} \in \mathcal{U}([-1,1])$, $i = 1,\ldots,m; j = 2,\ldots,n+1$ (input weights). Here, m denotes the number of basis functions. These choices correspond to the original suggestions given in the ELM portal [45], although using only $\mathcal{U}([-1,1])$ is currently the most common choice [46]. Let $\mathbf{H} = \left[\mathbf{h}_i\right]_{i=1}^N \in \mathbb{R}^{m\times N}$ be the matrix representation of the generated random basis.

In the basic ELM [9, 10], one directly solves the weight matrix $\mathbf{W} \in \mathbb{R}^{k\times m}$ for the linear combination of the generated basis from the identity $\mathbf{Y} \simeq \mathbf{W}\mathbf{H}$, i.e.,

$$\mathbf{W} = \mathbf{Y}\mathbf{H}^+,$$

where + denotes the pseudoinverse of $\mathbf{H}$. Note that, as argued in the work of Huang [37] and Huang et al. [19], the basic form of the ELM—or any other technique here—does not include the bias in the hidden layer. However, our reason for this choice is not generally based on the same arguments as in Huang [37] but on the results from Corollary 1 in Kärkkäinen [47], Kärkkäinen and Heikkola [48]: *With the hidden layer bias, one always obtains a model with mean error over the training data equal to zero.* Even if such a condition statistically guarantees an unbiased nonlinear regression and classification model, the zero mean error is also a constraint that remains always valid for all random feature mappings. We chose to avoid this for larger data-based flexibility of the models. Actually, the explicit condition and the corresponding constraint just described provide concrete closure of the corresponding discussion in Huang [37, p. 384] when referring to [49]: "existence of bias b may result in additional constraints and make the final solution tend to be suboptimal".

For the formulations in this paper, let us turn our attention to the other main form of learning in the ELM by considering the regularized least-squares optimization problem [19, 32, 47, 48]:

$$\min_{\mathbf{V}\in\mathbb{R}^{k\times m}} \mathcal{J}(\mathbf{V}), \tag{2}$$

where

$$\mathcal{J}(\mathbf{V}) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathbf{V}\mathbf{h}_i - \mathbf{y}_i\|_2^2 + \frac{\alpha}{2m}\sum_{i=1}^{k}\sum_{j=1}^{m} |\mathbf{V}_{ij}|^2. \tag{3}$$

The coefficients $\frac{1}{N}$ and $\frac{1}{m}$ in $\mathcal{J}(\mathbf{V})$ balance the scales of the fidelity and the regularization terms with respect to the amount of data and the size of the basis, respectively [50]. In (3), $\alpha > 0$ is the Tykhonov regularization/weight decay/shrinkage/penalization parameter, which restricts the increase in the magnitude of the weights and by enforcing strict coercivity, guarantees the unique solvability of (3). This technique is supported by the results in Bartlett [51], where it was shown that a large feedforward network with small training set error should favor small weights for improved generalization. However, we will use a very small α, so that the computational stability and uniqueness are the essential reasons to use the regularization technique here.

The solution $\mathbf{W} \in \mathbb{R}^{k\times m}$ of (2), i.e., the unique minimizer of (3), satisfies

$$\frac{1}{N}(\mathbf{W}\mathbf{H} - \mathbf{Y})\mathbf{H}^T + \frac{\alpha}{m}\mathbf{W} = \mathbf{0}, \tag{4}$$

so it can be solved from

$$\mathbf{W}\left(\mathbf{H}\mathbf{H}^T + \frac{\alpha N}{m}\mathbf{I}\right) = \mathbf{Y}\mathbf{H}^T, \tag{5}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$ denotes the identity matrix.

To this end, for the ELM, the class of an unseen input vector $\tilde{\mathbf{x}}$ is given by the maximum component $\max_j \mathbf{o}_j$ of the k-dimensional output-vector $\mathbf{o} = \mathbf{W}\tilde{\mathbf{h}}$, where $\tilde{\mathbf{h}}$ is computed according to (1).

2.2. The MLM

The learning method of the minimal learning machine is composed of the two main phases [21, 22]:

1. Construction of the distance-based regression model and,
2. Estimation of the distance-regression based output of an unseen test input.

For the first phase of the original MLM, the construction of distance-based random feature mapping, let us select m reference points $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^m$ such that, for all i, $\mathbf{r}_i = \mathbf{x}_j$ for some $1 \le j \le N$. Hence, $\mathbf{R}$ is a random subset of the set of input vectors. Let also $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^m$ refer to the outputs of the corresponding reference inputs, i.e., $\mathbf{r}_i \mapsto \mathbf{t}_i$ for all $1 \le i \le m$ in the training data.

Using the set of reference points and the whole set of input vectors, define the distance matrix $\mathbf{H} \in \mathbb{R}^{m\times N}$ as

$$(\mathbf{H})_{ij} = \|\mathbf{r}_i - \mathbf{x}_j\|_2, \quad i = 1,\ldots,m,\ j = 1,\ldots,N. \tag{6}$$

Similarly, define the output distance matrix $\mathbf{D}_y \in \mathbb{R}^{m\times N}$ as

$$\mathbf{D}_y = \left[\|\mathbf{t}_i - \mathbf{y}_j\|\right] \quad i = 1,\ldots,m,\ j = 1,\ldots,N. \tag{7}$$

The principal assumption in the first step of the MLM is the existence of a regression model between the distance matrices: $\mathbf{D}_y = g(\mathbf{H}) + E$, where E denotes the residual error and g the regression model. Assuming further that g is linear allows one to represent it as a matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$, which can be estimated in a similar fashion as in (4)–(5) using regularized ordinary-least-squares [36, 52]:

$$\mathbf{B} = \left(\mathbf{H}\mathbf{H}^T + \alpha\mathbf{I}\right)^{-1}\mathbf{H}\mathbf{D}_y^T. \tag{8}$$

Again, $\alpha > 0$ guarantees the unique solvability of (8) because the outer-product matrix $\mathbf{H}\mathbf{H}^T$ is always at least positive semidefinite [53].

For the derivation of the second step in the original MLM, let $\tilde{\mathbf{x}}$ be an unseen input vector whose MLM-output is to be estimated. The output of the first step gives the distance vector $\delta^T \in \mathbb{R}^m$ satisfying the identity

$$\delta = \left[\|\tilde{\mathbf{x}} - \mathbf{r}_i\|\right]_{i=1}^{m}\mathbf{B}.$$

These distances are then used to define the multilateration problem [23, 25], again in the form of a least-squares problem:

$$\tilde{\mathbf{y}}^* = \operatorname{argmin}\mathcal{J}(\tilde{\mathbf{y}}),$$

where

$$\mathcal{J}(\tilde{\mathbf{y}}) = \sum_{i=1}^{m}\left(\|\tilde{\mathbf{y}} - \mathbf{t}_i\|^2 - \delta_i^2\right)^2. \tag{9}$$

The minimizer of (9) provides the output vector of the MLM, whose maximum component determines the class label of $\tilde{\mathbf{x}}$. As described in [22], one can apply many nonlinear optimization solvers in (9). Moreover, the second-order Newton's method with a special initialization strategy was suggested and experimented in [36].

As far as the author is aware, no universal approximation results for the MLM exist. Compared to the ELM, there are two main differences in the construction, which makes analysis of the universal approximation capability also different from the ELM. Namely, a) the distance-based basis computed during the first phase of the MLM is not random in the sense of the random generation of $\{\mathbf{w}_i, b_i\}$ for the ELM [15, Definition 11.2]; b) the set of reference points is selected from the training set and therefore, can only provide information available there and not from a continuous sampling distribution as in the ELM. This information is 'nonlinearized' in the MLM through the lift to the distance regression, whose results are used to interpolate the actual MLM output vector and the corresponding class label during the second phase. For universal approximation, the role of both phases in the construction of the MLM output should be understood and analyzed.

Because we confine ourselves to the classification problems here, we eventually choose to use the efficient *fast MLM* nearest neighbor-based solution method in the second phase of the MLM, as described in the work of Mesquita et al. [35, Section 3.1]. Instead of minimizing (9) for the *1-of-k* coding scheme, Mesquita et al. [35] showed that it is sufficient to search the minimum component of the distance vector δ and recover the label of the corresponding output reference point. Since its introduction, the fast MLM has been favorably compared to many other techniques in the works of Marinho et al. [30, 31]. The overall fast MLM classification method with the training and application phases are depicted in Algorithms 1 and 2.

Algorithm 1 *TrainMLM* - Training phase of the MLM.

Input: Input-output-class label data $\{\mathbf{x}_i, \mathbf{y}_i, l_i\}_{i=1}^N$, number of reference points m, and regularization parameter α.

Output: Set of reference points $\{\mathbf{r}_i\}_{i=1}^m$ and their class labels $\{l_i\}_{i=1}^m$, distance regression weights $\mathbf{B} \in \mathbb{R}^{m\times m}$.

1. Select m reference points $\{\mathbf{r}_i\}_{i=1}^m$ from $\mathbf{X}$ and store corresponding labels $\{l_i\}_{i=1}^m$
2. Compute $\mathbf{H}$ from (6) and $\mathbf{D}_y$ from (7)
3. Solve $\mathbf{B}$ from (8)

Algorithm 2 *ApplyMLM* - Classification phase of the MLM.

Input: Reference points $\{\mathbf{r}_i\}_{i=1}^m$, reference labels $\{l_i\}_{i=1}^m$, weight matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$, and a set of new inputs $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^M$

Output: Set of labels $\{\tilde{l}_i\}_{i=1}^M$ for $\tilde{\mathbf{X}}$

1. Compute the distance regression matrix $\tilde{\mathbf{H}} = \mathbf{B}\mathbf{A} \in \mathbb{R}^{m\times M}$, where
$$(\mathbf{A})_{ij} = \|\mathbf{r}_i - \tilde{\mathbf{x}}_j\|_2, \quad i = 1,\ldots,m,\ j = 1,\ldots,M$$
2. Seek the minimum indices $\mathbf{J}_i = \operatorname{argmin}_{1\le j\le m}(\tilde{\mathbf{h}}_i)_j$
3. Set $\tilde{l}_j = l(\mathbf{J}_j)$, $j = 1,\ldots,M$

In relation to the ELM, the computational costs of the training phase of the MLM are comparable: We need to create two matrices and solve one linear problem whereas the sigmoidal activation needs to be computed and a similar $m \times m$ linear problem solved with the ELM. However, the fast MLM needs more memory compared to the ELM, because of the $m \times m$ matrix $\mathbf{B}$ compared to the $k \times m$ matrix $\mathbf{W}$. Moreover, the second phase of the original MLM with the repeated solution of the multilateration problem is computationally more involved compared to the ELM, and so is the fast MLM: It contains larger matrix computations and searches the best indices during Step 2 of Algorithm 2 from m-dimensional vectors instead of the k-dimensional outputs of the ELM. In summary, even the fast MLM will be a computationally more expensive classifier compared to the ELM.

Algorithm 3 *TrainEMLM* - Training phase of the EMLM.

Input: Input-output data $\mathbf{X} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, number of reference points m, and regularization parameter α.
Output: Set of reference points $\{\mathbf{r}_i\}_{i=1}^m$ and output weights $\mathbf{W} \in \mathbb{R}^{k \times m}$

1. Select m reference points $\{\mathbf{r}_i\}_{i=1}^m$ from $\mathbf{X}$
2. Compute $\mathbf{H}$ using formula (6)
3. Solve $\mathbf{W}$ from (5)

Algorithm 4 *ApplyEMLM* - Classification phase of the EMLM.

Input: Reference points $\{\mathbf{r}_i\}_{i=1}^m$, weight matrix $\mathbf{W} \in \mathbb{R}^{k \times m}$, and a set of new inputs $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^M$
Output: Set of labels $\{l_i\}_{i=1}^M$ for $\tilde{\mathbf{X}}$

1. Compute the distance matrix $\tilde{\mathbf{H}} \in \mathbb{R}^{m \times M}$ as

$$(\tilde{\mathbf{H}})_{ij} = \|\mathbf{r}_i - \tilde{\mathbf{x}}_j\|_2, \quad i = 1, \ldots, m, \;\; j = 1, \ldots, M$$

2. $l_i = \text{argmax}_{1 \le j \le k} (\mathbf{o}_i)_j \quad \text{for} \quad \mathbf{o}_i = \mathbf{W}\tilde{\mathbf{h}}_i$

2.3. The EMLM

A method referred to as the *extreme minimal learning machine (EMLM)* is obtained when distance-based feature mapping (6) is used with regularized ELM satisfying (5). The two main algorithms of the resulting method, training and classification of new instances, are depicted in Algorithms 3 and 4. Note that in practice, with a training data with inputs and their labels, the class encoding with the *1-of-k* coding scheme for the output matrix $\mathbf{Y}$ should be computed in the beginning of Algorithm 3. All three methods need such a step, so it will have no effect on the comparison of the computational costs.

Let us comment on the proposed method. The basic ingredient is that the sigmoidal transformation of input vectors in the ELM is replaced with the distance-based feature mapping underlying the MLM. The only metaparameter, m (when confined to the Euclidean distance; also other choices are possible [22, 30, 31]), refers to the number of reference points. Compared to the ELM and the MLM, the nonlinearity in (6) is not based on any transformation with a nonlinear function as in the ELM or the lift to the level of distance regression as in the MLM. Thus, it will be interesting to experiment with the consequences of this choice in the learning capability of the new method (cf. [54]).

The components of the derived EMLM method are close to those of the ELM with the multiquadric activation function, because of the input-vector distance calculations. However, these two methods are not the same: With the ELM, we have random generation of the hidden weights $\{\mathbf{a}_i, b_i\}$, and in the EMLM, $b_i = 0$ and $\{\mathbf{a}_i\}$'s correspond to the chosen set of reference points. Moreover, similarly to the ELM [e.g., 37, formula (10)], the solution to the least-squares learning problem in the EMLM without hidden-layer bias coincides with the least-squares support vector machine (SVM) to solve the Lagrange multipliers [55]. Also along the lines of the terminology popularized by the SVM, computation of the distance matrix in (6) is realization of a similarity function, a kernel [56], using the selected set of reference points. Hence, the kernel trick of the EMLM is to use the distance in the original space and not to visit a higher dimensional space as in the SVM.

As can be seen from Algorithms 3 and 4, the computational costs of the EMLM with random selection of the reference points are close to those of the ELM. The second stage, i.e., actual classification of test data, should be faster for the EMLM compared to the fast MLM because less storage is needed for the data structures to be passed from the training phase to the application phase. Again similarly to the ELM, much lower-dimensional vectors need to be processed with the EMLM to detect the final class compared to the MLM. In essence, the storage and computational costs of the basic EMLM outperform those of the MLM.

Selection of the reference inputs for the MLM in regression problems, by using the clustering initialization algorithm from Gonzalez [57], was suggested and experimented in [36]. This method, referred as *RS-maxmin*, provides a completely deterministic selection strategy of the reference points. Namely, the observation closest to the data mean is chosen as the first reference point. Then, a new observation that has the largest distance to the already chosen set of reference points is added to this set until the number of reference points, m, is reached. To carry out a more versatile comparison of the ELM, the MLM, and the EMLM, we choose to use the RS-maxmin for the reference point selection with the EMLM. Methodologically, the purpose is to stabilize further the randomness of the EMLM and assess the properties of the resulting algorithm. Use of RS-maxmin will introduce additional computational costs in Step 1 of Algorithm 3.

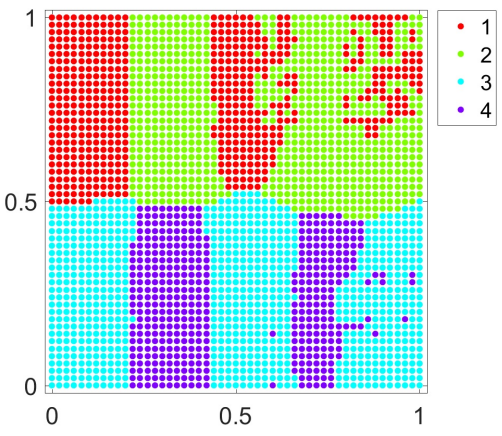

Fig. 1. Illustration of the EMLM with the Overlap dataset.

Remark 1. *The structure of the EMLM—linear combination of the distance-based feature mapping—coincides with the basic form of the radial basis function network (RBFN) with linear kernel [58, 59]. Also the use of the regularized least-squares learning framework to determine the weights was mentioned in [58], although not in the form as defined in* (3)*. For RBFN, centers that correspond to the reference points according to the MLM terminology are typically selected randomly or by using clustering, usually k-means [60]. In this respect, the use of RS-maxmin is both novel and computationally more efficient.*

The structural correspondence between the EMLM and RBFN means that the universal approximation properties proved to the latter model, e.g., in [58, 59, 61], are also valid to the EMLM. Hence, in combination with the results presented in Sections 2.1 and 2.2, we conclude that both ELM and EMLM posses universal approximation capability, which has not been established for MLM.

To this end, the capability of creating disjoint and nonlinear class boundaries using the EMLM technique is illustrated in Fig. 1. Using the training set of the 2-dimensional dataset 'Overlap' (see Table 1), after min-max scaling into $[0, 1]$, we first applied Algorithm 3 with 1000 reference points and the RS-maxmin selection. Then Algorithm 4 was applied for the test data covering $[0, 1] \times [0, 1]$ uniformly with grid size $h = \frac{1}{50}$. In Fig. 1, the class labels are illustrated with different colors.

3. Experimental results

Reference versions of the techniques in Section 2 were implemented with Matlab (version R2015b). As explained, the output vectors were formed using 1-of-k encoding, and we selected $\alpha = \sqrt{\varepsilon}$, where ε is the machine epsilon (of the order 10^{-16}), was fixed throughout. In preprocessing, we removed the constant variables and min-max scaled all features into $[0, 1]$. Compared to the initial assessments presented in [62], the stability problems in training for larger values of m with the MLM and the EMLM were omitted by using left division and Gaussian elimination instead of forming the explicit inverse for solving (5) and (8) (note that left division and explicit inversion are included as options by Huang [45]).

The datasets for the tests mostly originate from the UCI machine learning repository [63]. Only datasets with the availability of an independent validation set were considered. Typically, the class frequencies in the training and validation sets were consistent, i.e., approximately of the same size ($\pm 0 - 3\%$). However, the 'CrowdSource' dataset [64] provides a severe exception with the training set class frequency percentages of [13.7 70.5 4.2 9.2 0.5 1.9], but the validation set frequency percentages [17.7 26.0 12.0 13.3 15.7 15.3]. Thus, higher accuracy in the validation set does not imply a better classifier *per se*. For proper treatments of such a discrepancy, one should apply the classwise weighting method in the learning problem as suggested by Kärkkäinen [47, Section 3.4.4] and Huang et al. [19, Section 4.4].

Table 1
Description of test datasets. (* = Constant input features removed)

Dataname	N	NV	n	k	R-MCP
COIL	1 800	5 400	20*	100	3.5 [66]
Outdoor	2 400	1 600	21	40	28.6 [66]
Optdigits	3 823	1 797	61*	10	2.0 [67]
Overlap	3 960	990	2	4	16.3 [66]
HumActRec	4 252	1 492	561	6	0.3 [68]
Satimage	4 435	2 000	36	6	11.0 [20]
USPS	7 291	2 007	256	10	4.4 [66]
Isolet	6 238	1 559	617*	26	3.3 [66]
CrowdSource	10 545	300	28	6	28.0 [64]
Letter	16 000	4 000	16	26	2.9 [66]
MNIST	60 000	10 000	666*	10	1.5 [66]

Because of the incremental flavor of the MLM and the EMLM and for comparison, we used many of the same datasets as in the works of Losing et al. [65, 66]. The datasets are described in Table 1. There, N denotes the number of training observations and NV the number of validation observations; n refers to the dimension of the input data vectors; k provides the number of classes; and 'R-MCP' presents a reference classification accuracy result as MisClassifications in Percentages (MCP) with a citation to the work from where it was retrieved. More specifically, the reference results from Losing et al. [66] are given according to Table 3/Setting 1, for 'Optdigits' from Alpaydin and Kaynak [67] (1NN), for 'HumActRec' from Davis et al. [68], for 'Satimage' from Ding et al. [20] based on Table 4, and for 'CrowdSource' from Johnson and Iizuka [64] using Table 4.

Note that the reference results cannot be treated as precise benchmark values because they have been obtained using different comparison frameworks with different error computation formulae (e.g., the mean error over 10 repetitions [66]). Classification results, in general and for random basis, can be improved with careful feature selection [69], class imbalance management [70], robustness to outliers [33, 48, 71], rigorous architecture design [72], classification task simplification using one-versus-all or one-versus-one binarization approaches [73–75] etc.; see the works of Huang et al. [19], Ding et al. [20]. These techniques were not used here, and therefore, the reference results simply provide some basic level of classification accuracy in the separate validation set.

The main goals of the experiments for comparisons of the three methods were to

i) evaluate the training set classification accuracies to assess and verify the discrete universal approximation properties (cf. [54]),

ii) study the generalization potential by investigating the

best validation set classification accuracies,

iii) consider the determination of the metaparameter m, and

iv) compare the experimental computational complexity using CPU times.

Concerning the last point, the experiments reported here were computed using many laptops and workstations with single processors (2.6–2.8 GHz). Therefore, the CPU times are comparable only individually and separately for each dataset and not between the different datasets or different scenarios described in Sections 3.1 and 3.2. Moreover, especially when comparing the ELM and the EMLM, we are essentially assessing the efficiency of the implementations of the *exp*- and *pdist2*-functions in Matlab.

In the tests, we applied an incremental strategy for the values of the metaparameter m by grid searching 1% to 100% portions of the training data. More precisely, the initial m was set to $\lfloor 0.01 \cdot N \rfloor$ and then incremented with the stepsize $\lfloor 0.01 \cdot N \rfloor$ up to, at most, N. In the cross-validation tests in Section 3.2, the largest value of m was restricted to the smallest size of the 10 training sets (c. $\lfloor 0.9 \cdot N \rfloor$).

Figs. 3–23 given at the end of the paper illustrate the experimental behavior of the methods for most of the datasets, omitting 'Outdoor', 'Satimage', 'USPS', and 'Letter'. Multiple figures are included because *i*) there is common behavior in the experiments for different datasets, but *ii*) the commonalities are not encapsulated on a dataset basis but on the overall experimental and comparative level.

3.1. Training and validation set accuracy with full training

We first explain the basic setting for the experiments with the full data and then state and discuss the results.

3.1.1. Setting

We focus on experimental goals *i*) and *ii*) by studying the universal approximation capability, i.e., how well the training set can be learned, and the relations between the training and validation set accuracies. The experimental results are summarized in Table 2, where 'm' denotes the value of the metaparameter. 'TrMCP' refers to the MCP error in the training set and 'VaMCP' in the validation set, respectively. For 'VaMCP', m corresponds to the smallest overall error, and for 'TrMCP', to the first value when the training set MCP-error was below 0.1 (i.e., more than 99.9% classification accuracy). If the latter condition was not reached during the training, then the minimum value and the corresponding size of m were reported. For the MLM and the EMLM, we also included in the column 'VaLst' the validation set MCP error for the last, largest model that utilizes the whole training data ($m = N$). Finally, 'CPU' presents the total computing time in seconds for training the corresponding model.

3.1.2. Results

One can conclude from Table 2 that both the MLM and the EMLM are able to learn to classify the training set accurately. This is not a trivial result for a random basis with a particular learning framework. It also underlines that the regularization of the least-squares problems as defined in (3) and (8) does not disable the universal approximation capability of the fast MLM or the EMLM. However, the training of the ELM failed for 'COIL', 'Outdoor', and 'Overlap', on the grounds of the best training and validation set errors. These three datasets all have fewer features than the number of target classes.

The behavior of the training and validation set MCP errors for 'COIL' and 'Overlap' are depicted in Figs. 3 and 9, respectively ('Outdoor' behaves visually similarly to 'COIL'). Qualitatively, Fig. 3 also illustrates the typical form of the exponential decay of the training error with small values of m for all methods. However, the smallest dimensional dataset 'Overlap' is also a significant exception for the MLM and the EMLM: It is the only case where the best validation set error was obtained for both methods (see Table 2) with a relatively small value of m, with a slight increase in the validation error for the larger values of m. Interestingly, this exception can be seen in the form of the decrease in the training error: After a knee point [76], a linear decrease instead of an exponential decay is visible.

As shown in Table 2 and in the corresponding figures that compare the training and validation set errors, except for 'Overlap', the large training set classification accuracy and the best validation set error level for the MLM and the EMLM are typically obtained with a large value of m, close to the maximum value N. Moreover, as can be seen by comparing columns 'VaMCP' and 'VaLst', the validation error level with the distance-based basis does not increase when m is increased. There is again a slight exception to this general behavior provided by 'CrowdSource', but as explained above, the incompatibility between the training and validation set characteristics preclude interpreting this as a real counterexample.

For the ELM, we witnessed in all other cases than 'Overlap' (where the ELM training failed) overlearning in the form of a clear increase in the validation error when m was increased sufficiently. With 'COIL' and especially with 'CrowdSource', this happened very early. Moreover, the best validation errors obtained with the ELM were always larger compared to the MLM and the EMLM, which were very close to each other in all reasonable cases. This does not mean that the MLM and the EMLM dominate the ELM as classification techniques, but that the ELM was the greediest technique in learning, needing a better architecture design to choose the size of the hidden layer m more accurately than in the experiments here.

When comparing the best and last validation set results with the MLM and the EMLM to the reference values in the last column of Table 1, one concludes that typically

similar error levels were obtained. For 'Optdigits' the result was better than the original 1NN results reported with the dataset itself in [67]. However, for 'HumActRec' the results obtained were much worse than reported in the work of Davis et al. [68]. In the original work the benefits of using a binary classifier (there, SVM) with one-versus-all (OVA) strategy were pointed out. Because of this, the potential of the OVA approach was also briefly experimented. The result obtained, for the binary problem to separate class 1 from the rest with 'HumActRec', was that the MLM and the EMLM scored only a 3.2% MCP error level in this case.

This last point (and the choice of $m = N$ for the MLM and the EMLM) was also studied further by using the larger human activity dataset from Anguita et al. [73] (see also [74, 75]) with $N = 7352, n = 561, k = 6$ training data, and $NV = 2947$ validation data. For this dataset, the training and validation set class frequencies were balanced and consistent. The benchmark MCP error rate as reported in the work of Anguita et al. [73] was 4.0%, obtained with OVA-SVM. When this original training data was used with $m = N$, the MLM and the EMLM reached only 13% MCP error. But again, for the binary problem to separate class 1 from the rest, the error rate with $m = N$ readily decreased into 1.9% for the MLM and the EMLM.

3.2. Metaparameter selection

We first explain the basic setting for the experiments with the cross-validation and then state and discuss the results.

3.2.1. Setting

We consider experimental goal *iii*) next. In the preliminary tests of the EMLM in [62], we used the leave-one-out cross-validation (LOO-CV) technique with the efficient TR-PRESS implementation [77] to identify the only metaparameter, m, needed in all the techniques. These experiments were not successful. Thus, instead of the LOO we tested the CV with the classical choice of 10 folds [78]. For the maximum similarity between the data subsets in folding, the *distribution optimally balanced stratified CV* (DOB-SCV) [79, 80] is applied, with the implementation described by Kärkkäinen [81].

The difficulties noted in [62] could also be due to the observations given in [81]: The cross-validation error, i.e., the mean over the test fold errors [34, 78], and the validation set error do not posses a high correlation for the discrete MCP error measure. Therefore, as suggested in [81], Mean-Root-Squared (MRS) error was used to compute the test errors in the folds. However, by construction, the fast MLM estimates only the labels in the test set, so that the cross-validation error for the MLM is still computed as the mean over the MCP errors in the folds.

Cross-validation results are given in Table 3. There, 'TsMRS' refers to the smallest MRS CV-error with the corresponding value of m. Similarly, 'VaMRS' with m indicates the smallest value of mean validation error, where for each m the MRS errors of the 10 different trained models on the whole validation set have been computed and averaged. The correlation coefficient 'Cor' between all 'TsMRS' and 'VaMRS', i.e., for all values of m is also reported. Again, 'CPU' presents the total computing time in seconds for training the individual models. In the cross-validation experiment, the dataset 'MNIST' was searched only up to half of the size of the training data.

3.2.2. Results

First, we notice that the exceptional learning behavior of the dataset 'Overlap', as discussed in Section 3.1, is also visible in Fig. 10 (right), which is the only case for the MLM CV error with a clear increase for larger values of m.

As with the LOO-CV in [62], the 10-fold CV had difficulty identifying the best model structure. These are best illustrated with the ELM which tends to overlearn. Except for 'Overlap' and 'Letter', also the CV error reflected this for the ELM with a clear increase for larger values of the size of the hidden layer. The values of m for the smallest validation error 'VaMCP' when the whole training data were explored in Table 2 and the two errors in Table 3, 'TsMRS' and 'VaMRS', for the ELM, are all different for all datasets. The differences in the values are so large that they cannot be explained with a coarse search grid of the values of m.

For the MLM, the best validation MCP errors in Table 2 and the CV MCP errors in Table 3 are very close. Most of the suggested values of m in Table 3 are large and consistent with Table 2, except for 'CrowdSource'. For the EMLM, this behavior is even more stable, so that 'TsMRS' and 'VaMRS' suggest and support the use of the largest possible m. In Fig. 11, we see that even for 'Overlap', the CV error and the mean validation error for the EMLM are smallest for the largest m (although the validation error increases in the middle range).

The smoothing effect of taking the mean of the test or validation set errors is illustrated in Tables 2 and 3 when comparing the 'VaMCP' and 'VaMRS' values for the stable techniques MLM and EMLM. With them, the CV-related error figures typically show decreasing trends for the CV and validation errors. With also the averaged validation error, the best m is high and always close to N. For COIL, the CV-error 'CVTs' for the MLM and the EMLM in Figs. 4 (right) and 5 is slightly larger, for 'Overlap' in Figs. 10 (right) and 11 about the same, for 'MNIST' with all three methods in Figs. 22 and 23 (left) exactly the same, but for all other datasets (cf. the corresponding figures) clearly smaller than the mean validation error 'CVVa'. The notable exception is 'Isolet', where the CV and validation MCP errors with the MLM coincide, as supported by the corresponding correlation coefficient in Table 3. Very large deviations between the testing and validation CV errors were witnessed for 'HumActRec', 'USPS', and 'Crowdsource', the last one as expected.

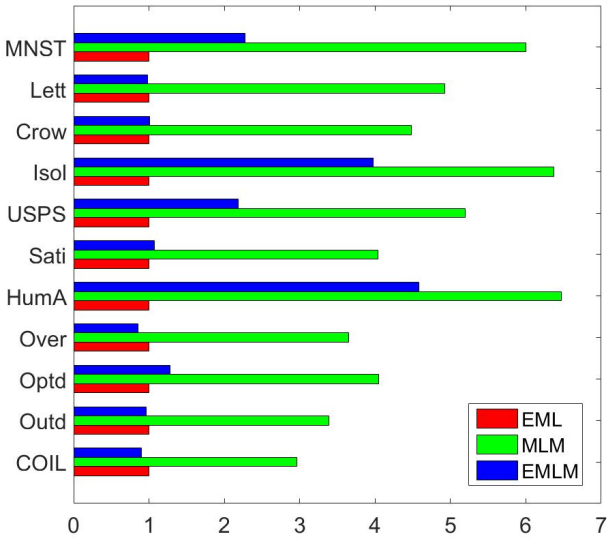

Fig. 2. Relative CPU times for all datasets.

3.3. Experimental computational complexity

Next, we consider and conclude the experiments in relation to goal *iv*). The computing times for the three methods are given in Tables 2 and 3 in column 'CPU'. There, the ELM is the fastest method to train, usually 2-5 times faster than the MLM. One cannot state a definite ranking between the training speeds of the MLM and the EMLM, as they are close to each other with varying order in individual datasets.

However, we can have two scenarios concerning the CPU time of the EMLM, when the RS-maxmin reference point selection method is used. In the tests reported thus far, the RS-maxmin algorithm was always run from the scratch. In this scenario, the experiments with different values of m are assumed to be completely independent from each other. Then, the CPU time for the EMLM, even with lower memory and computational requirements compared to the MLM as explained in Section 2, does not fully capitalize this potential advantage. However, if the experimental scenario corresponded to the incremental search of the value of m, then also the deterministic RS-maxmin—sorting indices of the observations for the reference point selection—could be realized as a one-shot method. This is obtained when running the whole algorithm only once for the whole data and then picking a subset of size m from this result for individual runs.

With the latter scenario just explained, we repeated the experiments in Table 2 (for MNIST only up to $N/2$ similarly to Table 3), including the RS-maxmin CPU time in the CPU time of the EMLM. This experiment was carried out on 64-bit Windows 10 Enterprise, with 2.8 GHz CPU and 32 GB RAM. The results are shown in the bar plot in Fig. 2 where the CPU time of the ELM has been normalized to unity by dividing the CPU times of all three methods by that of the ELM for each dataset individually. This experiment confirms that the ELM is the fastest method to train. The EMLM can be 2–4 times slower than the ELM for datasets with a large number of features, affecting the costs of the distance computation, but of similar computational complexity for smaller-dimensional problems. The MLM is the most expensive classification technique, taking 3–6 times more CPU time compared to the ELM.

However, the computing time and the complexity of learning are put into a completely different perspective when we notice the results from the previous section: The EMLM especially does not need a grid search of m because the whole training data can be used for the distance-based regression model without overlearning. The CPU times of this approach can be compared to those of the ELM, when the ELM can reach a comparable validation set classification accuracy in Table 2. More precisely, with 'Optdigits' (for $m = 1365$) and 'USPS' (for $m = 2044$) the ELM obtained a VaMPC error similar to the EMLM with the whole data. Thus, we repeated these experiments with a restricted maximum for the ELM and the whole data for the EMLM. We obtained the following results:

1. 'Optdigits' ($m = 39 - 1365$ for the ELM): ELM-CPU = 4.6 (VaMPC = 1.6), EMLM-CPU = 6.1 (VaMPC = 1.2)
2. 'USPS' ($m = 73 - 2044$ for the ELM): ELM-CPU = 13 (VaMPC = 5.3), EMLM-CPU = 87 (VaMPC = 4.4)

This small experiment concluded that the ELM with a grid search is computationally more efficient than the EMLM with the whole data. As expected, the performance ratio becomes larger for larger datasets. However, the values of VaMPC obtained in this single experiment also illustrate the variability of the results for the ELM (when there was no repeated sampling as suggested in [16, 17]): We did not obtain as good VaMPC values now as in the original tests reported in Table 2. This prevents us from presenting definite conclusions about the computational efficiencies of the ELM and the EMLM.

3.4. Summary of the experimental results

In all tests, we noticed smoother behavior of the error curves for the EMLM compared to the MLM (and especially to the ELM). Compared to the MLM, this was because of the use of the stable and deterministic method RS-maxmin for selecting the reference points. Moreover, based on the correlation coefficients in Table 3, the EMLM was the most consistent technique by means of the relation with the CV and the mean validation error; only for 'Overlap' and 'CrowdSource' was the correlation coefficient not unity by two decimal places. In this direction, the ELM is the most unstable of the techniques, scoring even two large negative correlations coefficients with 'COIL' and 'Outdoor'.

With the CV error, the EMLM always outperformed the ELM by scoring smaller MRS error values, as shown in Table 3. However, the comparisons showed that for any of

the techniques, the 10-fold DOB-SCV was not completely reliable for estimating the generalization error as witnessed in a separate validation set. Thus, this result should not be judged exactly. However, the best mean validation errors for the EMLM were better than those of the ELM. But again, especially for larger and erroneous datasets, this reflects also the sparsity of the search grid for the values of m for the ELM.

'COIL', 'Outdoor', 'Overlap', and 'Letter' are the four datasets with more classes than input features. For these datasets, the results were not as good as the reference results given in Table 1. In addition, we observed many forms of inconsistent behavior in the experiments as depicted in Figs. 3, 4, and 5 for 'COIL' and in Figs. 9, 10, and 11 for 'Overlap'. Note that in these cases one cannot guarantee that the unknown class separation mapping from $\mathbb{R}^n$ to $\mathbb{R}^k$ is surjective or, therefore, bijective. This seems to negatively affect to all techniques tested here. Thus, incremental and active learning techniques focusing on the necessary subset of observations, as thoroughly experimented in the work of Losing et al. [66], seem better choices for such problems.

With 'Optdigits' and 'Satimage', we obtained better results than that for the reference. The reference for 'Optdigits' was given with a very basic 1NN technique but for 'Satimage' with the ELM. However, with 'HumActRec' the results were much worse than the reference result. But the reason was probably the advantage of using the one-versus-all approach with an ensemble of binary classifiers. In addition, for 'CrowdSource', the reference results were much better than the ones here, but conclusions with this dataset should be made cautiously (or completely omitted) because of the class frequency incompatibility between the training and validation sets. Such cases call for transfer learning based approaches [82]. The error figures for these datasets are illustrated in Figs. 6, 7, and 8 ('Optdigits'); 12, 13, and 14 ('HumActRec'); and 18, 19, and 20 ('CrowdSource').

For the larger image classification datasets 'USPS', 'Isolet', and 'MNIST', with hundreds of features to separate some tens of classes, we obtained results that either agreed with or were slightly better than the reference results given in Table 1. Such cases seem to be a good setting for random basis techniques. The error figures for these datasets are illustrated in Figs. 15, 16, and 17 ('Isolet') and 21, 22, and 23 ('MNIST').

4. Conclusions and future work

In this work, a combination of two scalable machine learning techniques, the extreme learning machine and the minimal learning machine, with randomly generated basis was proposed. The straightforward idea was to use the distance-based feature mapping from the MLM in an ELM-like regularized least-squares learning framework. According to the original nomenclature, the proposed method was referred as the extreme minimal learning machine, EMLM.

Results indicate that the distance-based random basis is a viable option for random feature mapping in regularized learning. The EMLM with the deterministic RS-maxmin selection of the reference points had a more stable learning curve compared to the ELM or the MLM. The pure random generation of basis without resampling implied some variability of the learning curves for the ELM and the MLM, but this did not prevent the convergence (cf. [16, 17]), especially for the MLM. For the ELM, we emphasize that the basic learning framework that was applied here provides only a reference: Better control of the generated feature mapping to act in the nonsaturated region [28, 40, 41] and rigorous architecture design [72, 83] would provide better performance for the technique.

Surprisingly, the majority of the experimental results, for the fast MLM and especially for the EMLM, suggest that there is no overlearning in training for these two techniques. Thus, the whole training data with $m = N$ is an appropriate choice of reference points without any other search or determination algorithm. This choice yields to a parameter-free machine learning method (with the Euclidean distance), making the training of the EMLM straightforward and very simple; see Algorithms 3 and 4.

As in the previous results [81], the accuracy of the approximation of the generalization error with a cross-validation technique, for the MCP error and the MRS error, was not perfect. We witnessed overestimation but mostly underestimation of the mean validation error by the mean test fold, i.e., the CV error. Because the folds in the CV were balanced using DOB-SCV, this may indicate, also for datasets other than 'CrowdSource', that the training and validation set characteristics, especially class distributions, are not completely compatible in the tested datasets, which originate from real measurements.

The fact that the underestimation happened for 'COIL', the only dataset with a clearly larger validation set compared to the training set, suggests that difference in the sizes of the training and validation sets might cause such a discrepancy. We tested this point further by re-running the cross-validation test for 'COIL' with the interchanged roles of the training and validation sets (i.e., taking $N = 5400$ and $NV = 1800$). The result of this test for the EMLM is illustrated in Fig. 5 (right). The mean validation error is approximated very accurately, and this held true for the MLM as well, with practically exact agreement (visualization not included). Another test in the same direction was done with 'MNIST', by using Dob-SCV to define a 20% sample of the original training data. The CV error and the mean validation error for this case are illustrated in Fig. 23 (right). We see similar behavior, overestimation of the mean validation error, as in the original test with 'COIL'. This result supports the hypothesis that the relation between the sizes of the training and validation sets affects the accuracy of the CV error. This indicates that further tests for the accuracy of the cross-validation tech-

nique should be carried out with different divisions into training and validation sets.

Many other directions exist for future research with the proposed techniques. With distance-based feature mapping, one could apply other shrinkage methods in addition to regularized least-squares (or ridge regression) to control the complexity of the linear combination of basis [34]. In this direction, sparsity favoring methods [50] could and should be applied and experimented. Moreover, one could test different combinations of the distance-based and sigmoidal transformations to construct "more vivid" random feature mappings. On one hand, a sequential version would be to apply the sigmoidal transformation to the distance matrix (with some scaling because of the non-negativity of distances). A parallel combination, on the other hand, would be similar to the structure of the ELM as proposed in the work of Akusok et al. [84]: to first extend the feature vectors (or some reduced combination, e.g., using the principal components [53], of them) with sigmoidal transformation and then use these enlargements together with the original features in distance-based regression calculations.

Acknowledgments

This work was supported by the Academy of Finland from the projects 311877 (Demo) and 315550 (HNP-AI). The constructive feedback from the reviewers, improving the contents and the presentation, is sincerely acknowledged. The author is also grateful to MSc Joonas Hämäläinen for his help in carrying out the experiments reported here.

References

[1] M. J. D. Powell, Radial basis function for multivariable interpolation: a review, in: Algorithms for Approximation, Clarendon Press, Oxford, 1987, pp. 143–167.

[2] D. S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321–355.

[3] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural computation 7 (1995) 219–269.

[4] T.-Y. Kwok, D.-Y. Yeung, Efficient cross-validation for feedforward neural networks, in: Proceedings of IEEE International Conference on Neural Networks, volume 5, pp. 2789–2794.

[5] S. McLoone, M. D. Brown, G. Irwin, A. Lightbody, A hybrid linear/nonlinear training algorithm for feedforward neural networks, IEEE Transactions on Neural Networks 9 (1998) 669–684.

[6] C. Eliasmith, C. H. Anderson, Developing and applying a toolkit from a general neurocomputational framework, Neurocomputing 26 (1999) 1013–1018.

[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[8] P. Angelov, A. Sperduti, Challenges in deep learning, in: 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 485–495.

[9] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, in: in Proceedings of International Joint Conference on Neural Networks (IJCNN2004), volume 2, pp. 985–990.

[10] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

[11] C. Gallicchio, J. D. Martin-Guerrero, A. Micheli, E. Soria-Olivas, Randomized machine learning approaches: Recent developments and challenges, in: 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2017), pp. 77–86.

[12] W. Cao, X. Wang, Z. Ming, J. Gao, A review on neural networks with random weights, Neurocomputing 275 (2018) 278–287.

[13] Y.-H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (1992) 76–79.

[14] W. F. Schmidt, M. A. Kraaijveld, R. P. Duin, Feedforward neural networks with random weights, in: Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on, IEEE, pp. 1–4.

[15] G.-B. Huang, L. Chen, C. K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (2006) 879–892.

[16] X. Liu, S. Lin, J. Fang, Z. Xu, Is extreme learning machine feasible? a theoretical assessment (part i), IEEE Transactions on Neural Networks and Learning Systems 26 (2015) 7–20.

[17] S. Lin, X. Liu, J. Fang, Z. Xu, Is extreme learning machine feasible? a theoretical assessment (part ii), IEEE Transactions on Neural Networks and Learning Systems 26 (2015) 21–34.

[18] J. Cao, Z. Lin, Extreme learning machines on high dimensional and large data applications: a survey, Mathematical Problems in Engineering 2015 (2015).

[19] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: A review, Neural Networks 61 (2015) 32–48.

[20] S. Ding, H. Zhao, Y. Zhang, X. Xu, R. Nie, Extreme learning machine: algorithm, theory and applications, Artificial Intelligence Review 44 (2015) 103–115.

[21] A. H. de Souza Junior, F. Corona, Y. Miche, A. Lendasse, G. A. Barreto, O. Simula, Minimal learning machine: A new distance-based method for supervised learning, in: International Work Conference on Artificial Neural Networks (IWANN'2013), Springer, pp. 408–416.

[22] A. H. de Souza Junior, F. Corona, G. A. Barreto, Y. Miche, A. Lendasse, Minimal learning machine: A novel supervised distance-based approach for regression and classification, Neurocomputing 164 (2015) 34–44.

[23] J. B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.

[24] T. F. Cox, M. A. A. Cox, Multidimensional scaling, Chapman and Hall/CRC press, 2 edition, 2000.

[25] Y. Takane, F. W. Young, J. De Leeuw, Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features, Psychometrika 42 (1977) 7–67.

[26] D. P. P. Mesquita, J. P. P. Gomes, A. H. de Souza Junior, A minimal learning machine for datasets with missing values, in: 22nd International Conference on Neural Information Processing - ICONIP 2015, pp. 565–572.

[27] D. P. Mesquita, J. P. P. Gomes, A. H. de Souza Junior, J. S. Nobre, Euclidean distance estimation in incomplete datasets, Neurocomputing 248 (2017) 11–18.

[28] P. Gastaldo, F. Bisio, C. Gianoglio, E. Ragusa, R. Zunino, Learning with similarity functions: a novel design for the extreme learning machine, Neurocomputing 261 (2017) 37–49.

[29] L. B. Marinho, A. H. de Souza Junior, P. P. Rebouças Filho, A new approach to human activity recognition using machine learning techniques, in: International Conference on Intelligent Systems Design and Applications, Springer, pp. 529–538.

[30] L. B. Marinho, J. S. Almeida, J. W. M. Souza, V. H. C. Albuquerque, P. P. Rebouças Filho, A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images, Expert Systems

with Applications 72 (2017) 1–17.
[31] L. B. Marinho, P. P. Rebouças Filho, J. S. Almeida, J. W. M. Souza, A. H. de Souza Junior, V. H. C. de Albuquerque, A novel mobile robot localization approach based on classification with rejection option using computer vision, Computers & Electrical Engineering 68 (2018) 26–43.
[32] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2012) 513–529.
[33] J. P. P. Gomes, D. P. M. A. L. Freire, A. H. de Souza Junior, T. Kärkkäinen, A robust minimal learning machine based on the M-estimator, in: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2017, pp. 383–388.
[34] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, volume 1, Springer series in statistics, New York, 2001.
[35] D. P. Mesquita, J. P. P. Gomes, A. H. de Souza Junior, Ensemble of efficient minimal learning machines for classification and regression, Neural Processing Letters 46 (2017) 751–766.
[36] J. Hämäläinen, T. Kärkkäinen, J. P. P. Gomes, Clustering-based reference points selection for the minimal learning machine, Manuscript, 2018.
[37] G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cognitive Computation 6 (2014) 376–390.
[38] G.-B. Huang, What are extreme learning machines? filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle, Cognitive Computation 7 (2015) 263–278.
[39] L. L. C. Kasun, Y. Yang, G.-B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Transactions on Image Processing 25 (2016) 3906–3918.
[40] J. Y. Yam, T. W. Chow, Feedforward networks training speed enhancement by optimal initialization of the synaptic coefficients, IEEE Transactions on Neural Networks 12 (2001) 430–434.
[41] G. Dudek, A method of generating random weights and biases in feedforward neural networks with random hidden nodes, arXiv preprint arXiv:1710.04874 (2017).
[42] A. N. Gorban, I. Y. Tyukin, D. V. Prokhorov, K. I. Sofeikov, Approximation with random bases: Pro et contra, Information Sciences 364 (2016) 129–145.
[43] M. Li, D. Wang, Insights into randomized algorithms for neural networks: Practical issues and common pitfalls, Information Sciences 382 (2017) 170–178.
[44] X. Liu, L. Xu, The universal consistency of extreme learning machine, Neurocomputing 311 (2018) 176–182.
[45] G.-B. Huang, MATLAB codes of ELM algorithm, 2013. `http://www.ntu.edu.sg/home/egbhuang/elm_random_hidden_nodes.html`.
[46] W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing 261 (2017) 28–36.
[47] T. Kärkkäinen, Mlp in layer-wise form with applications to weight decay, Neural Computation 14 (2002) 1451–1480.
[48] T. Kärkkäinen, E. Heikkola, Robust formulations for training multilayer perceptrons, Neural Computation 16 (2004) 837–862.
[49] J. A. Suykens, T. Van Gestel, J. De Brabanter, Least squares support vector machines, World Scientific, 2002.
[50] T. Kärkkäinen, R. Glowinski, A Douglas-Rachford method for sparse Extreme Learning Machine, Methods and Applications of Analysis (2018) 1–17. (in review).
[51] P. L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, IEEE Transactions on Information Theory 44 (1998) 525–536.
[52] A. S. Alencar, W. L. Caldas, J. P. Gomes, A. H. de Souza, P. A. Aguilar, C. Rodrigues, W. Franco, M. F. de Castro, R. M. Andrade, MLM-rank: A ranking algorithm based on the minimal learning machine, in: Brazilian Conference on Intelligent Systems (BRACIS-2015), IEEE, pp. 305–309.
[53] T. Kärkkäinen, M. Saarela, Robust principal component analysis of data with missing values, in: Lecture Notes in Artificial Intelligence (9166), Springer International Publishing, 2015, pp. 140–154.
[54] G.-B. Huang, Learning capability and storage capacity of two-hidden-layer feedforward networks, IEEE Transactions on Neural Networks 14 (2003) 274–281.
[55] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (1999) 293–300.
[56] M. J. Zaki, W. Meira Jr, W. Meira, Data mining and analysis: fundamental concepts and algorithms, Cambridge University Press, 2014.
[57] T. F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 38 (1985) 293–306.
[58] T. Poggio, F. Girosi, Networks for approximation and learning, Proceedings of the IEEE 78 (1990) 1481–1497.
[59] J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural computation 3 (1991) 246–257.
[60] S. Haykin, Neural Networks and Learning Machines, Prentice Hall, third edition, 2009.
[61] Y. Liao, S.-C. Fang, H. L. Nuttle, Relaxed conditions for radial-basis function networks to be universal approximators, Neural Networks 16 (2003) 1019–1028.
[62] T. Kärkkäinen, Extreme minimal learning machine, in: 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2018, pp. 237–242.
[63] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, 2017. http://archive.ics.uci.edu/ml.
[64] B. A. Johnson, K. Iizuka, Integrating openstreetmap crowdsourced data and landsat time-series imagery for rapid land use/land cover (lulc) mapping: Case study of the laguna de bay area of the philippines, Applied Geography 67 (2016) 140–149.
[65] V. Losing, B. Hammer, H. Wersing, Choosing the best algorithm for an incremental on-line learning task, in: 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 369–374.
[66] V. Losing, B. Hammer, H. Wersing, Incremental on-line learning: A review and comparison of state of the art algorithms, Neurocomputing 275 (2018) 1261–1274.
[67] E. Alpaydin, C. Kaynak, Optical recognition of handwritten digits, 1998. https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/optdigits.names.
[68] K. Davis, E. Owusu, V. Bastani, L. Marcenaro, J. Hu, C. Regazzoni, L. Feijs, Activity recognition based on inertial sensors for ambient assisted living, in: Information Fusion (FUSION), 2016 19th International Conference on, IEEE, pp. 371–378.
[69] Y.-P. Zhao, Y.-T. Pan, F.-Q. Song, L. Sun, T.-H. Chen, Feature selection of generalized extreme learning machine for regression problems, Neurocomputing 275 (2018) 2810–2823.
[70] W. Xiao, J. Zhang, Y. Li, S. Zhang, W. Yang, Class-specific cost regulation extreme learning machine for imbalanced classification, Neurocomputing 261 (2017) 70–82.
[71] K. Zhang, M. Luo, Outlier-robust extreme learning machine for regression problems, Neurocomputing 151 (2015) 1519–1527.
[72] Y. Sun, Y. Chen, Y. Yuan, G. Wang, Dynamic adjustment of hidden layer structure for convex incremental extreme learning machine, Neurocomputing 261 (2017) 83–93.
[73] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, in: 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), pp. 437–442.
[74] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in: International workshop on ambient assisted living, Springer, pp. 216–223.
[75] D. Anguita, A. Ghio, L. Oneto, F. X. Llanas Parra, J. L. Reyes Ortiz, Energy efficient smartphone-based activity recognition using fixed-point arithmetic, Journal of universal com-

puter science 19 (2013) 1295–1314.
[76] R. L. Thorndike, Who belongs in the family?, Psychometrika 18 (1953) 267–276.
[77] Y. Miche, M. Van Heeswijk, P. Bas, O. Simula, A. Lendasse, TROP-ELM: a double-regularized ELM using LARS and tikhonov regularization, Neurocomputing 74 (2011) 2413–2421.
[78] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95), volume 2, pp. 1137–1145.
[79] J. G. Moreno-Torres, J. A. Sáez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Transactions on Neural Networks and Learning Systems 23 (2012) 1304–1312.
[80] V. López, A. Fernández, F. Herrera, On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed, Information Sciences 257 (2014) 1–13.
[81] T. Kärkkäinen, On cross-validation for MLP model evaluation, in: Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science (8621), Springer-Verlag, 2014, pp. 291–300.
[82] S. M. Salaken, A. Khosravi, T. Nguyen, S. Nahavandi, Extreme learning machine based transfer learning algorithms: A survey, Neurocomputing 267 (2017) 516–524.
[83] P. H. Kassani, A. B. J. Teoh, E. Kim, Sparse pseudoinverse incremental extreme learning machine, Neurocomputing 287 (2018) 128–142.
[84] A. Akusok, K.-M. Björk, Y. Miche, A. Lendasse, High-performance extreme learning machines: a complete toolbox for big data applications, IEEE Access 3 (2015) 1011–1025.

Table 3
Cross-validation results.

	ELM				MLM				EMLM			
Dataname	*m*: TsMRS	*m*: VaMRS	Cor	CPU	*m*: TsMCP	*m*: VaMCP	Cor	CPU	*m*: TsMRS	*m*: VaMRS	Cor	CPU
COIL	486: 5.89e-1	72: 8.14e-1	-0.52	1.42e2	1242: 8.60e0	1548: 4.64e0	1.00	4.03e2	1584: 2.25e-1	1584: 1.69e-1	1.00	2.09e2
Outdoor	816: 5.17e-1	24: 8.95e-1	-0.54	1.59e2	1800: 2.92e0	2016: 3.06e1	0.98	5.30e2	2160: 1.71e-1	2136: 5.43e-1	1.00	2.87e2
Optdigits	1092: 2.66e-1	1053: 3.10e-1	1.00	5.15e2	2964: 7.57e-1	3393: 1.34e0	0.98	3.27e3	3432: 1.57e-1	3432: 2.01e-1	1.00	2.24e3
Overlap	3560: 5.54e-1	360: 5.50e-1	0.81	4.68e2	480: 1.68e1	600: 1.69e1	0.93	1.72e3	3120: 3.01e-1	3440: 3.02e-1	0.97	5.98e2
HumActRec	860: 4.68e-1	473: 6.18e-1	0.99	8.14e2	3698: 5.03e0	3784: 1.29e1	0.99	6.71e3	3784: 2.88e-1	3784: 4.00e-1	1.00	2.65e5
Satimage	405: 3.66e-1	225: 4.23e-1	1.00	6.80e2	3600: 6.63e0	3645: 8.77e0	0.97	2.72e3	3960: 2.07e-1	3915: 2.54e-1	1.00	1.91e3
USPS	1533: 2.83e-1	1533: 3.27e-1	1.00	3.01e3	5986: 1.44e0	6424: 4.56e0	0.99	4.26e4	6497: 1.61e-1	6424: 2.06e-1	1.00	3.76e4
Isolet	1071: 5.88e-1	819: 6.44e-1	1.00	2.09e3	5166: 2.66e0	5481: 3.14e0	1.00	1.46e4	5607: 3.31e-1	5607: 3.56e-1	1.00	6.07e4
CrowdSource	1908: 2.56e-1	212: 7.54e-1	0.98	6.50e3	9434: 2.73e0	3392: 3.82e1	0.69	2.88e4	9734: 1.39e-1	9328: 6.29e-1	0.94	1.30e4
Letter	8320: 3.84e-1	3520: 5.72e-1	0.36	2.54e4	14080: 2.46e0	14240: 4.48e0	1.00	1.17e5	14240: 2.19e-1	14240: 3.09e-1	1.00	3.44e4
MNIST	12e3: 3.14e-1	11400: 3.13e-1	1.00	2.56e5	28800: 1.71e0	29400: 1.75e0	1.00	1.62e6	29400: 1.92e-1	29400: 1.94e-1	1.00	3.40e6

Table 2
Complete data training results.

	ELM			MLM				EMLM			
Dataname	*m*: TrMCP	*m*: VaMCP	CPU	*m*: TrMCP	*m*: VaMCP	VaLst	CPU	*m*: TrMCP	*m*: VaMCP	VaLst	CPU
COIL	1782: 4.8	144: 28.3	1.91e1	1764: 0.1	1764: 4.1	4.1	5.64e1	1746: 0.1	1800: 4.1	4.1	2.80e1
Outdoor	2376: 2.0	96: 42.3	2.12e1	2064: 0.1	1728: 29.4	30.4	7.09e1	2352: 0.1	1056: 29.9	30.4	3.77e1
Optdigits	936: 0.1	1365: 1.3	6.98e1	2964: 0.1	2184: 1.2	1.2	2.79e2	936: 0.1	1677: 1.2	1.2	3.01e2
Overlap	120: 27.8	200: 27.9	6.99e1	3960: 0.1	560: 16.3	18.5	2.64e2	3960: 0.1	800: 16.4	18.5	8.85e1
HumActRec	2365: 0.1	903: 17.3	9.78e1	3999: 0.1	3870: 12.4	12.4	6.63e2	3741: 0.1	4042: 12.4	12.5	2.98e3
Satimage	3870: 0.1	540: 11.8	9.83e1	4320: 0.1	3375: 8.3	8.8	3.91e2	4185: 0.1	3825: 8.3	8.6	2.75e2
USPS	1825: 0.1	2044: 4.5	3.75e2	6351: 0.1	6351: 4.3	4.4	1.96e3	4088: 0.1	5183: 4.3	4.4	4.21e3
Isolet	2646: 0.1	2016: 6.4	2.78e2	5355: 0.1	4662: 2.9	3.0	1.80e3	5544: 0.1	5922: 3.0	3.0	7.10e3
CrowdSource	8692: 0.1	106: 40.0	9.93e2	9964: 0.1	1802: 36.7	40.3	4.41e3	9752: 0.1	2014: 38.0	40.0	1.73e3
Letter	12800: 0.1	3040: 7.3	3.16e3	15040: 0.1	14560: 4.2	4.2	1.57e4	11040: 0.1	13600:4.2	4.2	4.08e3
MNIST	18000: 0.1	21600: 1.9	1.29e4	54600: 0.1	52200: 1.5	1.6	6.31e4	39600: 0.1	37200: 1.5	1.6	5.63e5

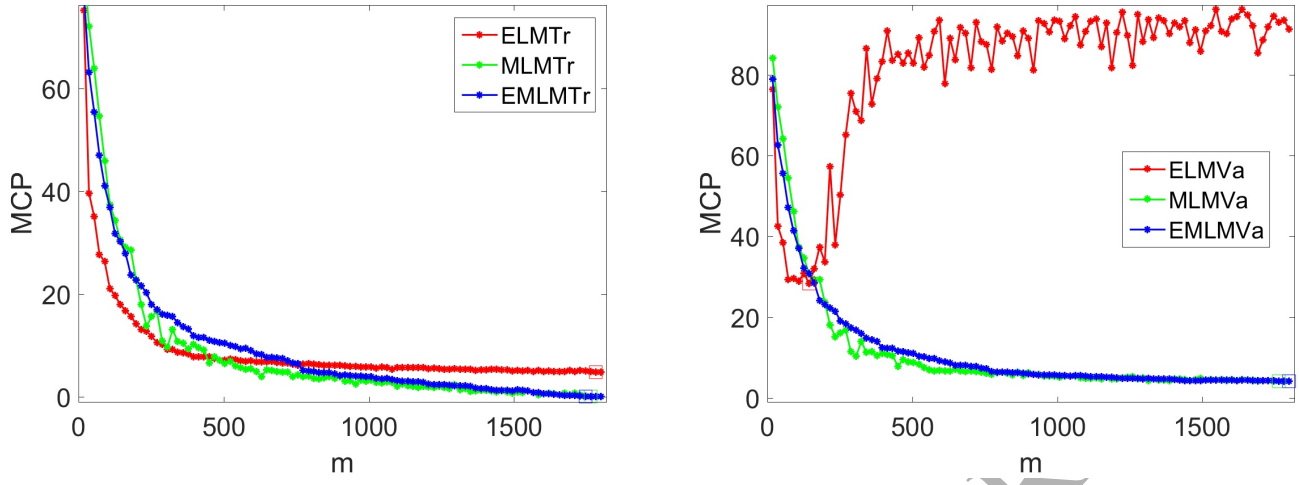

Fig. 3. COIL: training errors (left) and validation errors (right).

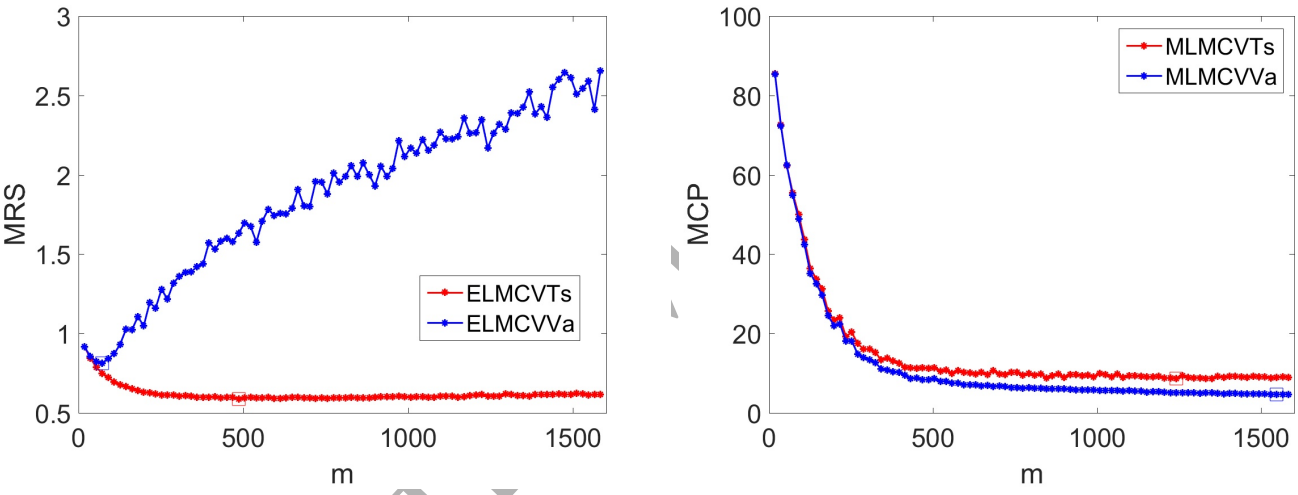

Fig. 4. COIL: cross-validation and mean validation errors for ELM (left) and MLM (right).

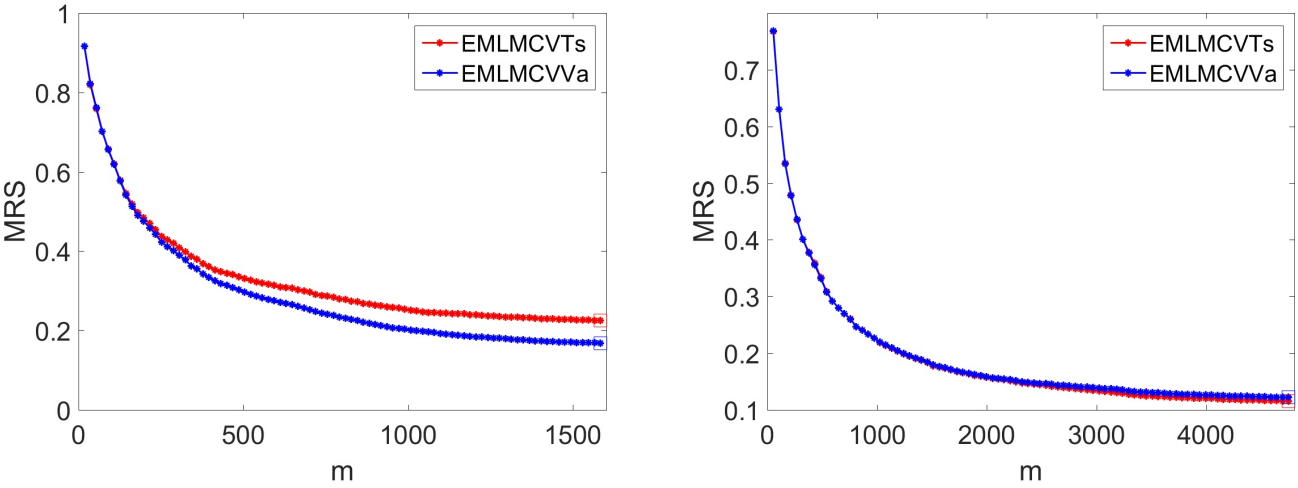

Fig. 5. COIL: cross-validation and mean validation errors for EMLM (left) and with training and validation set interchanged (right).

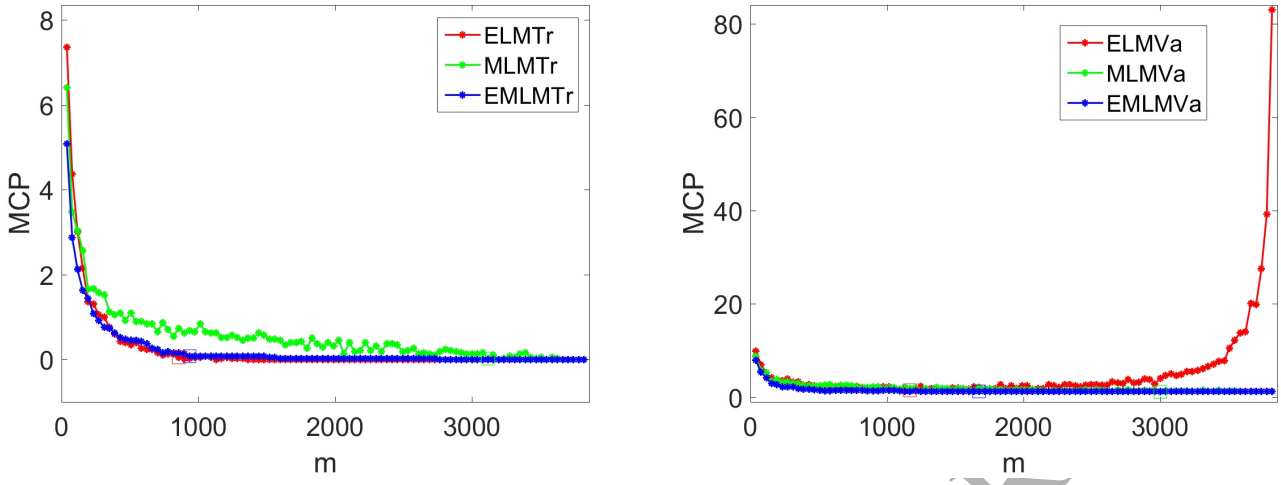

Fig. 6. Optdigits: training errors (left) and validation errors (right).

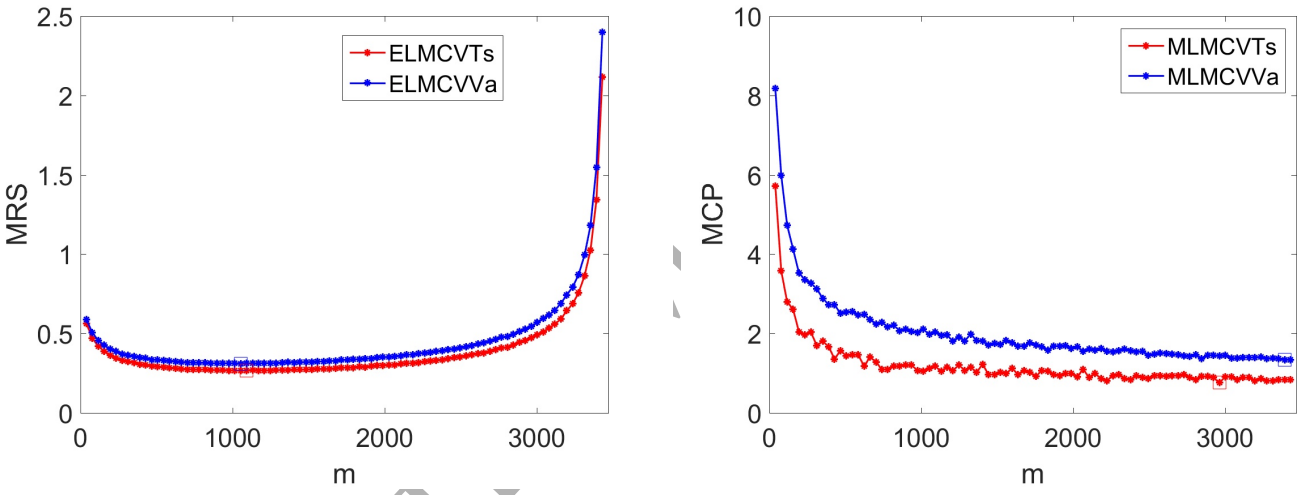

Fig. 7. Optdigits: cross-validation and mean validation errors for ELM (left) and MLM (right).

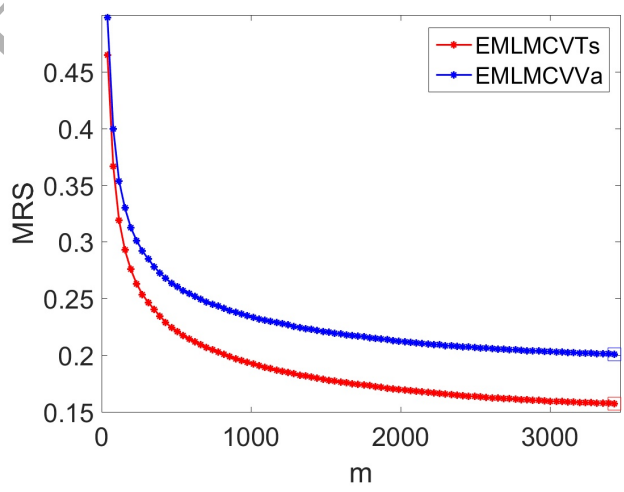

Fig. 8. Optdigits: cross-validation and mean validation errors for EMLM.

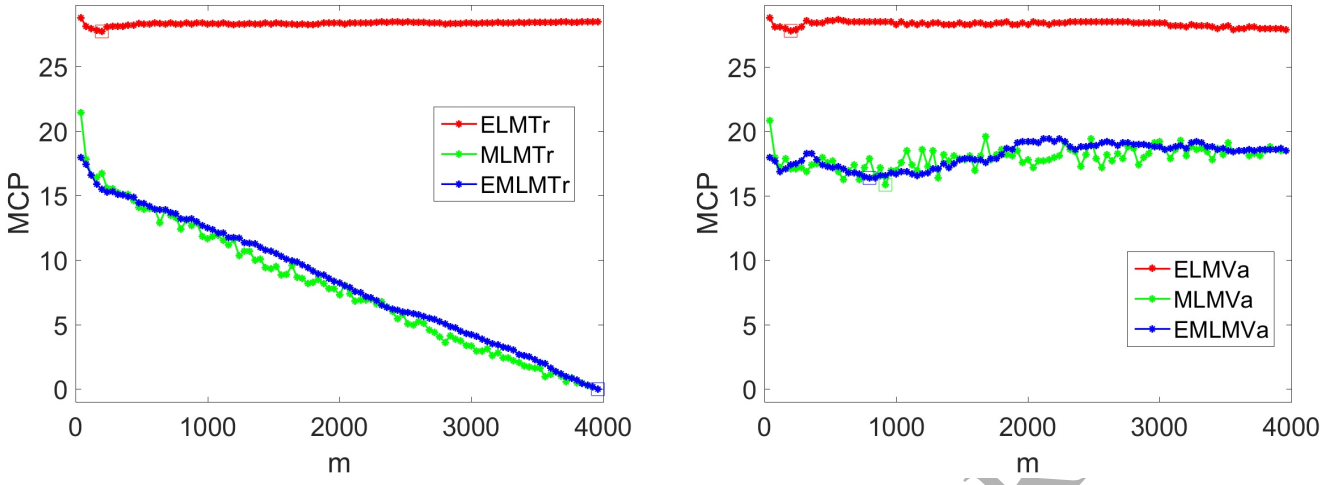

Fig. 9. Overlap: training errors (left) and validation errors (right).

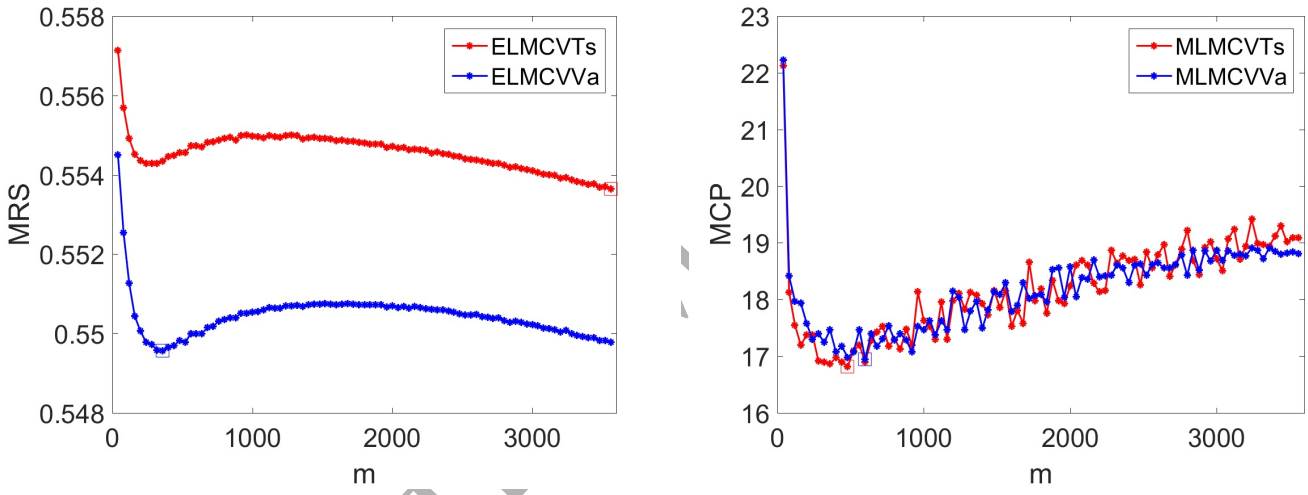

Fig. 10. Overlap: cross-validation and mean validation errors for ELM (left) and MLM (right).

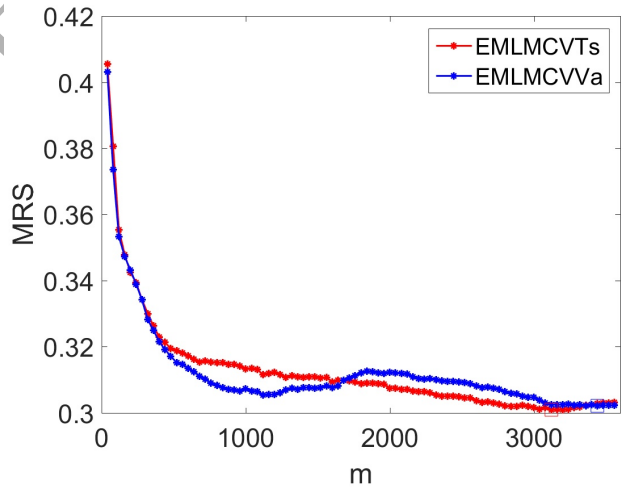

Fig. 11. Overlap: cross-validation and mean validation errors for EMLM.

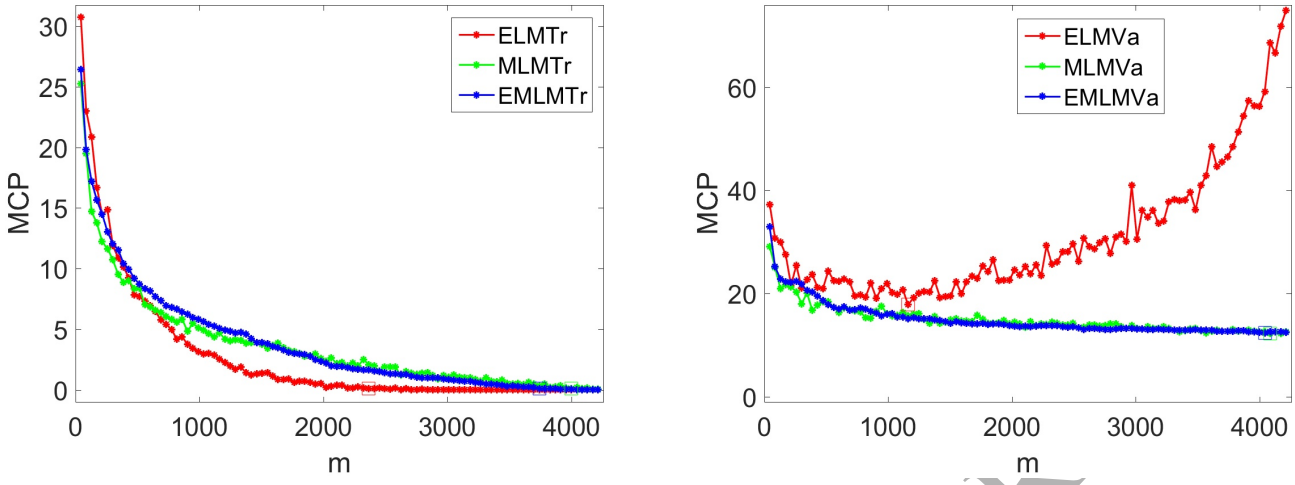

Fig. 12. HumActRec: training errors (left) and validation errors (right).

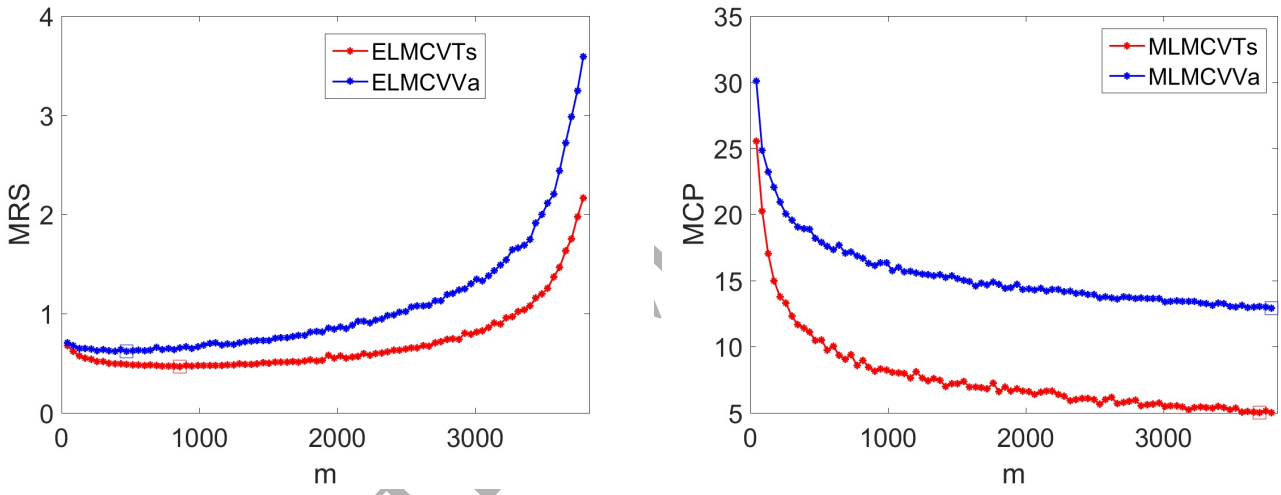

Fig. 13. HumActRec: cross-validation and mean validation errors for ELM (left) and MLM (right).

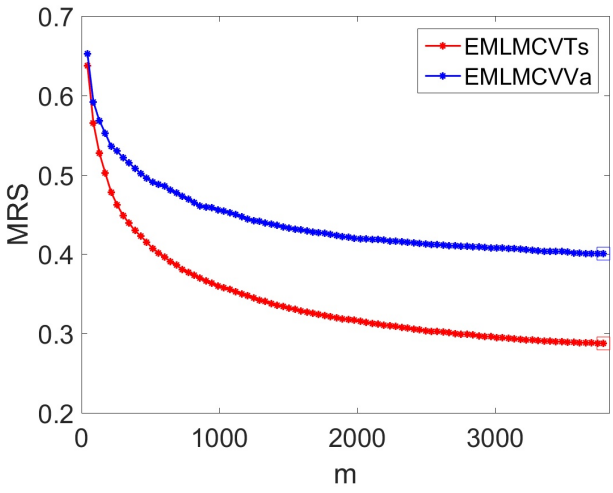

Fig. 14. HumActRec: cross-validation and mean validation errors for EMLM.

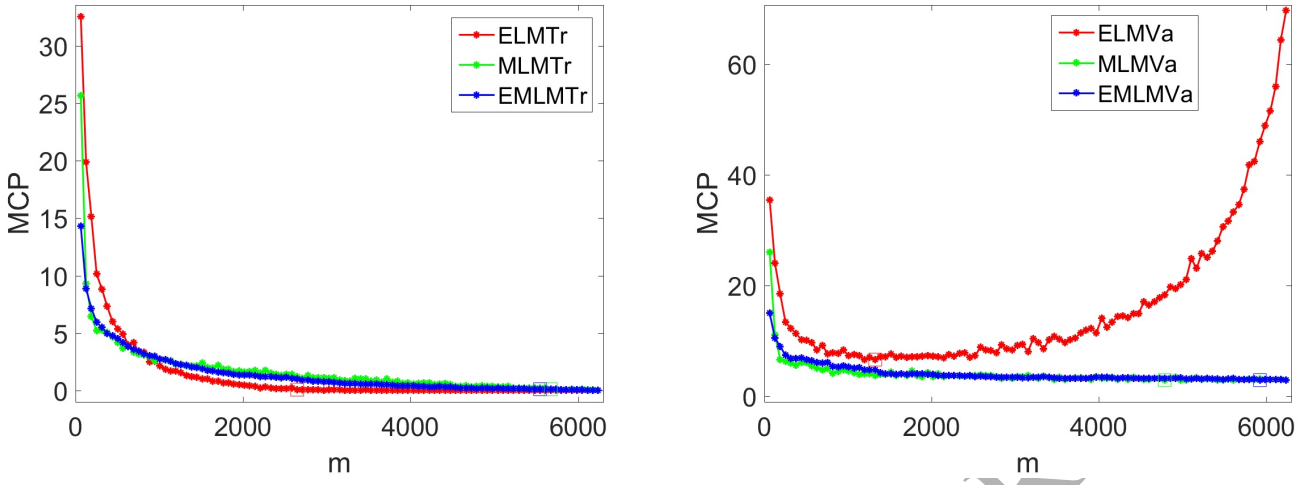

Fig. 15. Isolet: training errors (left) and validation errors (right).

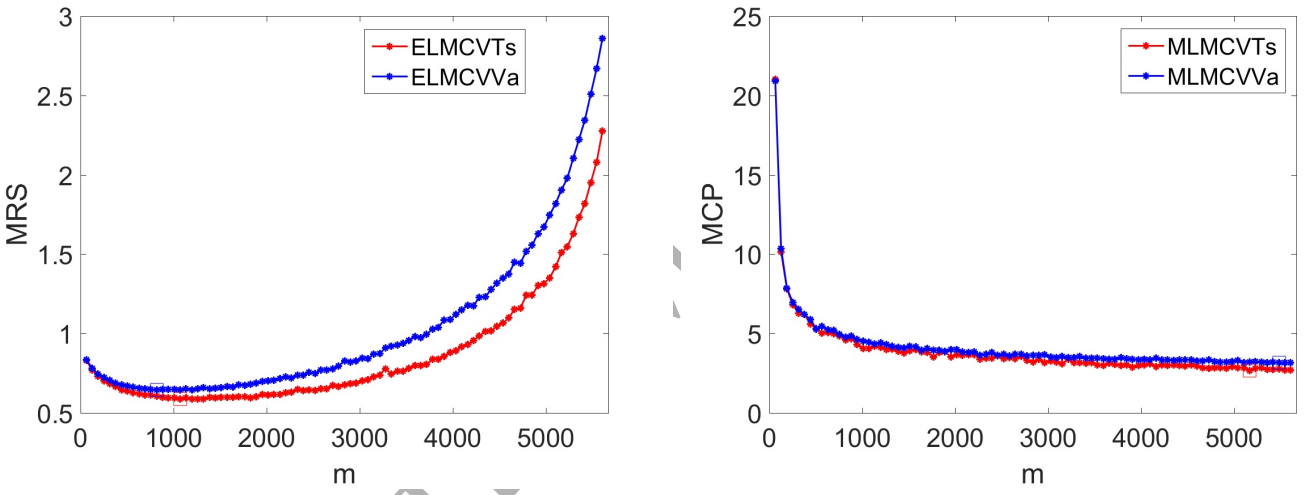

Fig. 16. Isolet: cross-validation and mean validation errors for ELM (left) and MLM (right).

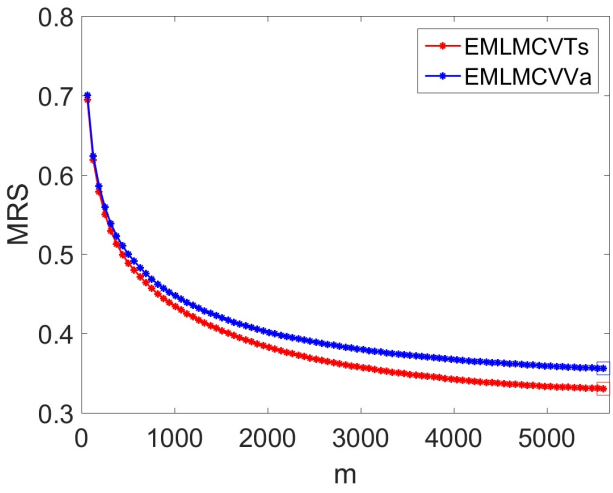

Fig. 17. Isolet: cross-validation and mean validation errors for EMLM.

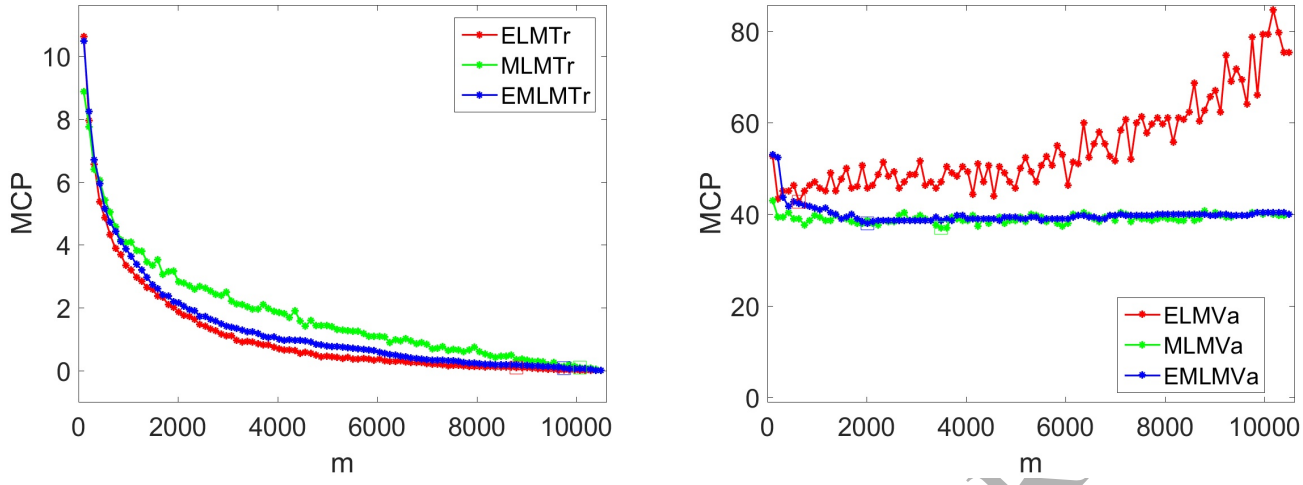

Fig. 18. CrowdSource: training errors (left) and validation errors (right).

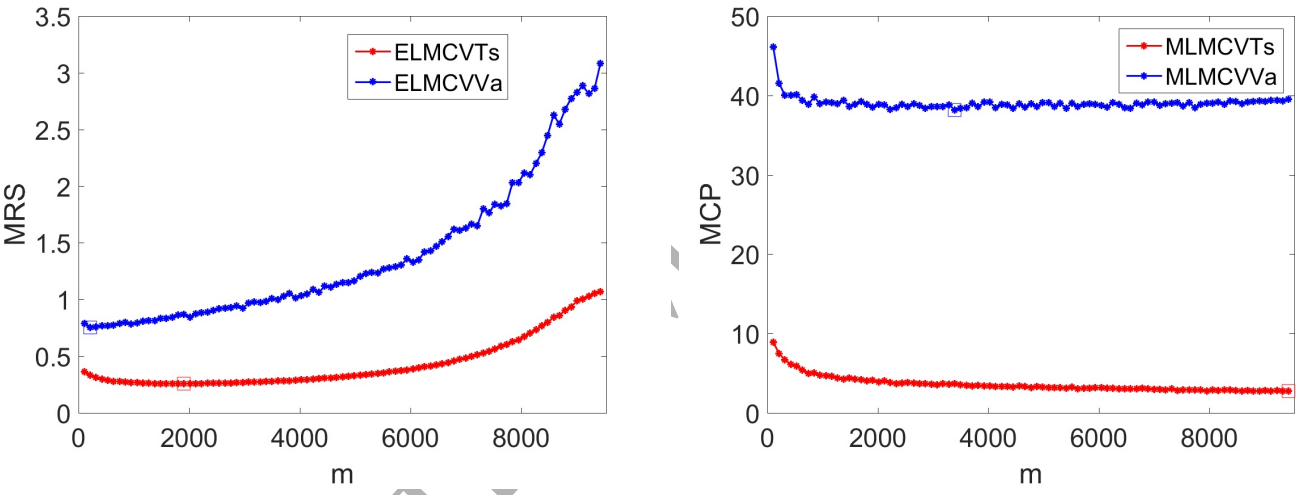

Fig. 19. CrowdSource: cross-validation and mean validation errors for ELM (left) and MLM (right).

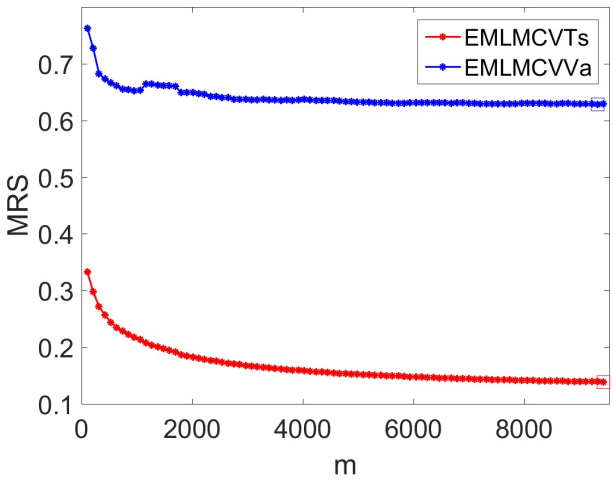

Fig. 20. CrowdSource: cross-validation and mean validation errors for EMLM.

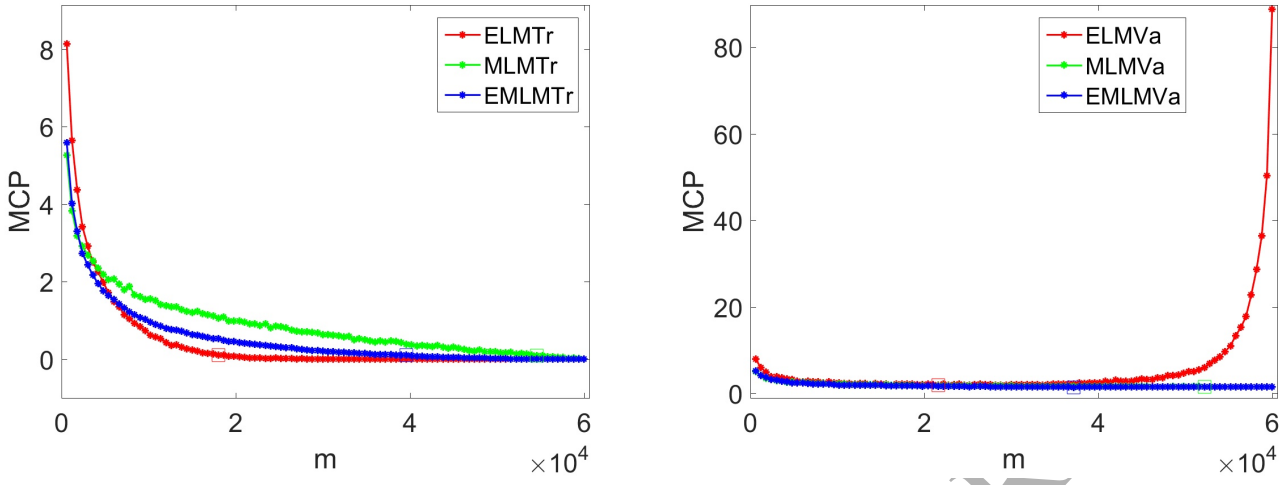

Fig. 21. MNIST: training errors (left) and validation errors (right).

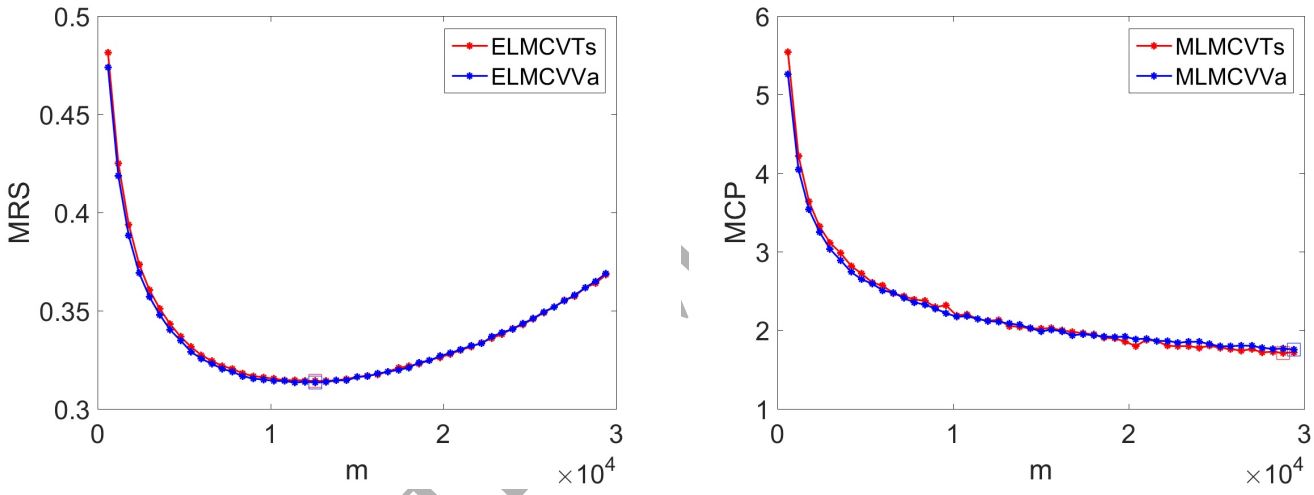

Fig. 22. MNIST: cross-validation and mean validation errors for ELM (left) and MLM (right).

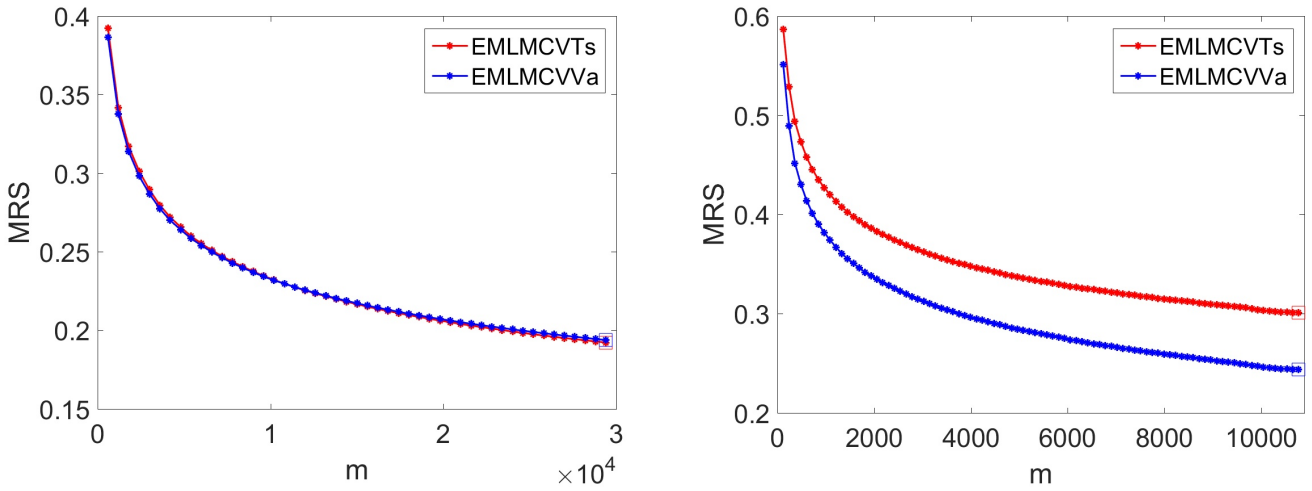

Fig. 23. MNIST: cross-validation and mean validation errors for EMLM: up to $N/2$ (left) and with 20% sample of $N = 12005$ (right).

Biography

Tommi Kärkkäinen completed his PhD in 1995 and he has worked as a full professor in the Faculty of Information Technology, University of Jyväskylä, since 2002. He has been and is serving in many positions of administration and responsibility at the faculty and the university level. His main research fields include computational sciences (optimization, data mining, machine learning) and computing education research. He has published over 180 research papers on various topics, led over 40 R&D projects, and supervised over 25 PhD theses.