ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Model selection for Extreme Minimal Learning Machine using sampling

Tommi Kärkkäinen

University of Jyvaskyla,
Faculty of Information Technology, Finland
tommi.karkkainen@jyu.fi

Abstract. A combination of Extreme Learning Machine (ELM) and Minimal Learning Machine (MLM)—to use a distance-based basis from MLM in the ridge regression like learning framework of ELM—was proposed in [8]. In the further experiments with the technique [9], it was concluded that in multilabel classification one can obtain a good validation error level without overlearning simply by using the whole training data for constructing the basis. Here, we consider possibilities to reduce the complexity of the resulting machine learning model, referred as the Extreme Minimal Leaning Machine (EMLM), by using a bidirectional sampling strategy: To sample both the feature space and the space of observations in order to identify a simpler EMLM without sacrificing its generalization performance.

1 Introduction

Sampling is a classical statistical strategy to reduce the number of samples and the amount of data processing [13]. In supervised learning, sampling can address either the set of observations or the feature space. Sampling of observations is usually related to model's generalization assessment using cross-validation [12, 6], also in connection to feature selection [7]. Generally forward or backward feature sampling and selection [14] is based on estimating relevances of the subsets of features. Random selection of features as a constituent of a machine learning method with integrated feature importance assessment strategy was popularized along with the Random Forest technique as proposed by Breiman [1].

Supervised training mechanisms that apply a random selection of constituents of a model can be traced back to [2]: Radial basis function networks, random vector functional link networks, Schmidt's method, and especially to Extreme Learning Machine [5]. A novel random method based on distances, the Minimal Learning Machine (MLM), was suggested and described in [3]. In the MLM, one creates a distance-based regression model between the input and output sampled distance matrices. After the distance-regression, MLM needs an additional multilateration step to interpolate a value of an unseen input point in the distance space. ELM and MLM were integrated in [8, 9] to *Extreme Minimal Learning Machine (EMLM)*, where the distance matrix from MLM was linked to the regularized least-squares learning framework characterizing ELM.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

The purpose of this paper is to elaborate on the model complexity of the novel technique EMLM. The starting point for this are the experimental results given in [9]: *the EMLM does not overlearn so a parameter-free supervised method is obtained by using the whole training data in learning*. We consider, through a new sampling-based algorithm that bidirectionally samples both input and feature space, whether we could simplify this full EMLM model withour loosing the generalization performance.

2 Methods and algorithms

Let a supervised training data with N observations be given: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^n$ (input) and $\mathbf{y}_i \in \mathbb{R}^k$ (output). In classification problems the given outputs $\mathbf{y}_i$ are formed using 1-of-k encoding with the class labels and we let $\mathbf{Y} = \left[\mathbf{y}_i\right]_{i=1}^N \in \mathbb{R}^{k\times N}$ be the matrix representations of these.

The first step in the MLM [3] contains computation of the following distance matrix $\mathbf{H} \in \mathbb{R}^{m\times N}$:

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

Here the dataset $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^m$ is referred as the set of *reference points*, which is sampled uniformly from the original set of inputs. In the Extreme Minimal Learning Machine (EMLM) [8, 9], this distance matrix is used as the feature map (kernel) in the regularized least-squares optimization problem [10]:

$$\min_{\mathbf{V}\in\mathbb{R}^{k\times m}} \mathcal{J}(\mathbf{V}), \text{ where } \mathcal{J}(\mathbf{V}) = \frac{1}{2N}\sum_{i=1}^{N}\|\mathbf{V}\mathbf{h}_i - \mathbf{y}_i\|_2^2 + \frac{\alpha}{2m}\sum_{i=1}^{k}\sum_{j=1}^{m}|\mathbf{V}_{ij}|^2. \tag{2}$$

The coefficients $\frac{1}{N}$ and $\frac{1}{m}$ in $\mathcal{J}(\mathbf{V})$ normalize the two terms and $\alpha > 0$ is the Tykhonov regularization/weight decay parameter, which, by enforcing strict coercivity, guarantees the unique solvability of (2). The solution $\mathbf{W} \in \mathbb{R}^{k\times m}$ of the least-squares problem satisfies the linear equation

$$\mathbf{W}(\mathbf{H}\mathbf{H}^T + \frac{\alpha N}{m}\mathbf{I}) = \mathbf{Y}\mathbf{H}^T. \tag{3}$$

Because the purpose of α is not to restrict the model complexity but just to assure the unique solvability of (3), we fix $\alpha = \sqrt{\varepsilon}$, where ε is the machine epsilon. The EMLM algorithms for training and application with an unseen test set are formalized in Algorithms 1 and 2. After Algorithm 2, the *misclassification-rate-in-percentages (MCP)* error of the EMLM for the test inputs is the relative amount of false labels in percentages.

As shown in Algorithm 2, complexity of an EMLM model is determined by the set of reference points in $\mathbb{R}^{m\times n}$ and the weight matrix $\mathbf{W} \in \mathbb{R}^{k\times m}$. A reduced model is the one which uses either smaller set of reference points m or less features n. For this purpose, we propose the following sampling-based model selection procedure for the EMLM:

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Algorithm 1 *TrainEMLM* - Training phase of the EMLM.

Input: Inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, outputs $\mathbf{Y}$, and number of reference points m
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 (1) and solve $\mathbf{W}$ from (3)

Algorithm 2 *ApplyEMLM* - Classification phase of the EMLM.

Input: Ref. points $\{\mathbf{r}_i\}_{i=1}^{m}$, weights $\mathbf{W} \in \mathbb{R}^{k \times m}$, and test 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 $\tilde{\mathbf{H}} = \left[\tilde{\mathbf{h}}_i\right] \in \mathbb{R}^{m \times M}$ as $(\tilde{\mathbf{H}})_{ij} = \|\mathbf{r}_i - \tilde{\mathbf{x}}_j\|_2$.
2. $l_i = \operatorname{argmax}_{1 \le j \le k} (\mathbf{o}_i)_j$ for $\mathbf{o}_i = \mathbf{W}\tilde{\mathbf{h}}_i$

EMLM Sampling Algorithm:

Step 0. Select two integer parameters $1 \le \mathit{Div} \le n$ and $1 \le \mathit{Nfld} \le N$. Set $\mathit{ModSiz} = \lceil n/\mathit{Div} \rceil$ and $\mathit{Nmodels} = \mathit{Nfld} \cdot n$.

Step 1. Divide the training data into *Nfld* folds using Dob-SCV algorithm [6].

Step 2. Loop from 1 up to *Nmodels* by doing

1. Select a fold from *Nfld* uniformly randomly.
2. Select *ModSiz* number of features from $\{1, \ldots, n\}$ uniformly randomly.
3. Train EMLM using Algorithm 1 with selected features and the fold as reference points. Estimate the accuracy of the model using Algorithm 2 in the whole training set with the selected features. Store the MCP error in two tables: one for the foldwise errors and one for the featurewise errors (copy of error for all features involved in the model construction).

Step 3. Compute mean foldwise *FoldErr* and mean featurewise *FeatErr* errors over the involved attempts. Sort these vectors in ascending order and divide both with their maximum values. Select thresholds $0 < \mathit{FoldThr} < 1$ and $0 < \mathit{FeatThr} < 1$. Remove folds with $\mathit{FoldErr} > \mathit{FoldThr}$ and features with $\mathit{FeatErr} > \mathit{FeatThr}$ from the final, reduced EMLM model.

Step 4. Train EMLM using Algorithm 1 with the selected folds and features. Estimate the accuracy of the model using Algorithm 2 with an unseen validation data.

Concerning the proposed algorithm, by using Dob-SCV we try to assure that all the folds approximate the data distribution of the input data [6, 7], so that only one of them can be used for training a reduced model—kind of dual strategy to cross-validation [6]. After sorting and division by maximum the last values of *FoldErr* and *FeatErr* equal to unity. Hence, with the proposed bounds for the thresholds we will reduce the EMLM model by omitting at least one fold and one feature using the sampling algorithm above.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Dataname	N	NV	n	k	FM/BM	RefCPU
Optdigits	3 823	1 797	61	10	1.2/1.2	3.0e2
Satimage	4 435	2 000	36	6	8.7/8.3	2.8e2
HumActRec	4 252	1 492	561	6	12.5/12.4	3.0e3
USPS	7 291	2 007	256	10	4.4/4.3	4.2e3
Isolet	6 238	1 559	617	26	3.0/3.0	7.1e3
MNIST	60 000	10 000	666	10	1.6/1.5	5.6e5
Gisette1	6 000	1 000	476	2	3.3/3.2	-
Gisette3	6 000	1 000	1 417	2	2.3/2.2	-
Gisette5	6 000	1 000	2 345	2	2.3/2.0	-

Table 1. Description of test datasets.

3 Experiments

Reference versions of the techniques in Section 2 were implemented with Matlab (R2015b), using the datasets described in Table 1. As preprocessing, we removed constant variables and min-max scaled all features into $[0, 1]$. We mostly use the same datasets as in [9] to enable comparison of the results to two MCP error base cases in a separate validation set of size NV: the full EMLM model's result and the best EMLM model's result along with the CPU time for searching the complete model [9] ('FM', 'BM', and 'RefCPU' in Table 1). The new dataset compared to [8, 9] is 'Gisette', because it contains half of true features and half of probe (noise) features by construction [4]. This dataset was sampled to create three additional test sets: 'Gisette1' with random selection of 10% of original features, 'Gisette3' with 30% and 'Gisette5' with 50% random sample of features.

Distance is a rigid concept and the euclidean distance underlines the statistical robustness [11]. Therefore, the original results with the distance-based basis are readily in a good level for the full EMLM in Table 1. This is especially seen with Gisette in Table 1, where half of trobe features did not harm EMLM notably when the number of included features was increased. Therefore, we chose the following thresholds: $FoldThr = 0.99$ and $FeatThr = [0.99\ 0.98\ 0.97\ 0.96]$, where after testing the latter sequence the model with the smallest training set error was selected. Because of the different characteristics of the datasets, *Div* and *Nfld* were fixed individually for each case. We tried to create strictly smaller models of size *ModSiz* ('MS') by selecting a large *Div*, still ensuring that the number of inputs for the reduced model was at least equal but typically strictly larger than the number of classes (see experiments and conclusions in [9]).

The model specifications and the results of the sampling-based model selection are given in Table 2. There, $\tilde{n}$ denotes the number of features in the reduced model and 'ValErr' the reduced model's MCP error in the validation set. 'Nmods' documents the total number of the reduced models tested during sampling and 'R%' gives the reduction rate of the final reduced EMLM model compared to the full model: $100m(k+\tilde{n})/(N(k+n))$. The column 'CPU' contains

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Data	*Div*	MS	*Nftd*	Nmods	m	$\tilde{n}$	R%	CPU	ValErr
Optdigits	5	12	8	488	1 918	49	42	3.0e1	1.3
Satimage	6	6	8	288	3 331	28	61	2.1e1	8.5
HumActRec	8	70	8	4 488	3 723	482	75	9.7e2	11.6
USPS	8	32	8	2 048	5 477	148	45	8.6e2	4.5
Isolet	8	77	8	4 936	4 679	578	71	2.7e3	2.9
MNIST	10	67	10	6 660	54 003	310	43	3.2e4	1.5
MNIST	10	67	20	13 320	51 007	444	57	2.9e4	1.5
Gisette1	8	60	8	3 808	3 750	181	24	1.4e3	2.8
Gisette3	15	94	8	11 336	5 250	543	34	1.2e4	2.1
Gisette5	15	156	8	18 760	3 000	456	10	4.8e4	2.0

Table 2. Experimental results.

the CPU time in seconds taken by the sampling algorithm, when one laptop and one workstation with 2.6–2.8 GHz processors were used in the experiments.

From Table 2 we conclude that the proposed sampling algorithm was able to reduce the complexity of the EMLM model while maintaining or slightly improving its independent validation accuracy. Tests with MNIST and Gisette show that the more we sample the more we are able to reduce the complexity of the EMLM. Comparison of 'CPU' to 'RefCPU' in Table 1 shows that the sampling procedure is faster than a full incremental search: the CPU times are 3 up to 18 (MNIST) times faster.

4 Conclusions

The purpose of the work was to study whether a less complex EMLM model with competitive classification accuracy could be obtained by sampling a batch of simpler models and selecting the reference points and features for the EMLM based on error samples. The proposed algorithmic skeleton provided the desired results: we were, indeed, able to identify a reduced model with similar or slightly better MCP error in the validation set compared to the full EMLM model or the best EMLM model from [9].

For some cases (e.g., USPS), the smallest training set error for the compared reduced models with different values of *FeatThr* did not give the smallest validation error. The observations in [6] suggest that one might be able to improve this by using a more sensitive error measure of the model's accuracy: The mean-root-squared-error could give better alignment with the training and validation set accuracies compared to the discrete MCP error that was applied here.

The results here document an initial assessment of the sampling strategy, which is easy to parallelize for larger problems. One could and should compare the selected features to the ones suggested by other techniques, notably Random Forest [1]. Perhaps uniform sampling of observations with smaller fold sizes would allow better separation of important and unnecessary reference points.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

One should also assess the selection of the metaparameters more thoroughly. This could be based on grid-search and cross-validation (CV) [7], although we encountered challenges with both LOO-CV and fold-based CV techniques with the random basis in [8, 9].

An interesting future direction would certainly be to derive a weighted distance measure based on the sampled accuracy of the features. Furthermore, sampling could be replaced or directed using analytical methods, because the simple form of the distance-based basis allows straightforward computation of the feature saliency similarly to [7].

Acknowledgments The work was supported by the Academy of Finland from the projects 311877 (Demo) and 315550 (HNP-AI).

References

1. L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
2. W. Cao, X. Wang, Z. Ming, and J. Gao. A review on neural networks with random weights. *Neurocomputing*, 275:278–287, 2018.
3. A. H. de Souza Junior, F. Corona, G. A. Barreto, Y. Miche, and A. Lendasse. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164:34–44, 2015.
4. I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider, and M. Uhr. Competitive baseline methods set new standards for the nips 2003 feature selection benchmark. *Pattern Recognition Letters*, 28(12):1438–1444, 2007.
5. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.
6. T. Kärkkäinen. On cross-validation for MLP model evaluation. In *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science (8621), pages 291–300. Springer-Verlag, 2014.
7. T. Kärkkäinen. Assessment of feature saliency of MLP using analytic sensitivity. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2015*, pages 273–278, 2015.
8. T. Kärkkäinen. Extreme Minimal Learning Machine. In *26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2018*, pages 237–242, 2018.
9. T. Kärkkäinen. Extreme Minimal Learning Machine: Ridge regression with distance-based basis. *Neurocomputing*, 2018. to appear, 21 pages.
10. T. Kärkkäinen and R. Glowinski. A Douglas-Rachford method for sparse Extreme Learning Machine. *Methods and Applications of Analysis*, pages 1–17, 2018. (in review).
11. T. Kärkkäinen and E. Heikkola. Robust formulations for training multilayer perceptrons. *Neural Computation*, 16(4):837–862, 2004.
12. 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, pages 1137–1145, 1995.
13. P. S. Levy and S. Lemeshow. *Sampling of populations: methods and applications*. John Wiley & Sons, 2013.
14. H. Liu and H. Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.