MDCC
Mestrado e Doutorado em Ciência da Computação

On Model Complexity Reduction in Instance-based Learners

Doctoral Program in Computer Science – Federal University of Ceará
Prof. Dr. João Paulo Pordeus Gomes (Advisor)
Prof. Dr. Ajalmar Rêgo da Rocha Neto (Co-advisor)

Ph.D. Candidate: **Saulo Anderson Freitas Oliveira**

May 12, 2021

Agenda

Introduction

Reducing Complexity By Instance Selection

- [Proposal#1] CCIS: Selecting class-corner samples
- [Proposal#2] CC-LSSVM: When LSSVM meets CCIS

Reducing Complexity by Instance Regularization

- [Proposal#3] LW-MLM: When MLM meets regularization by sample
- [Proposal#4] CCLW-MLM: When LW-MLM meets CCIS

Concluding Remarks

References

Introduction

Introduction

Background

- A large class of machine learning problems ends up as being equivalent to a function estimation/approximation task by digging in the information that resides in the available training data set.
- Focusing on the task of supervised learning which states the following:
  - Given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ of N example input–output sample pairs, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^S$ where each $\mathbf{y}_i$ was generated by an unknown function $\mathbf{y} = f(\mathbf{x})$, discover a function $h(\cdot)$ that approximates the true function $f(\cdot)$ (NORVIG; RUSSELL, 2017). Here, $h(\cdot)$ is selected from a hypothesis space $\mathcal{H}$.
- From the above, one can notice that learning is a search through the space of possible hypotheses for one that will perform well, even on new examples beyond the training set.

Introduction

Background

- The functional dependence that relates the input to the output:
  - **parametric modeling** *vs.* **nonparametric modeling**;
- Nonparametric modeling is also called **instance-based learning** or memory-based learning (THEODORIDIS, 2020);
- Instead of making explicit generalizations, they compare instances of new problems with instances seen in the learning process:
  - K-Nearest Neighbors (K-NN) (COVER; HART, 1967);
  - Support Vector Machine Machines (SVM) (CORTES; VAPNIK, 1995);
  - Relevance Vector Machine (RVM) (TIPPING, 2001); and, more recently,
  - Minimal Learning Machine (MLM) (JUNIOR *et al.*, 2015).

Introduction

The catch. . .

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

Introduction

The catch...

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

The Challenge...

- In this case, what means reducing complexity?

Introduction

The catch...

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

The Challenge...

- In this case, what means reducing complexity?
  Short Answer: Control the influence of samples during both training and prediction.
  - How to define a subset with K samples from the dataset with N samples ($N \geq K$)?
  - How to identify which specific combination of samples among the $\binom{N}{K}$ possible combinations?
    Spoiler: NP-Hard!

Introduction

The catch...

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

The Challenge...

- In this case, what means reducing complexity?
  Short Answer: Control the influence of samples during both training and prediction.
  - How to define a subset with K samples from the dataset with N samples ($N \geq K$)?
  - How to identify which specific combination of samples among the $\binom{N}{K}$ possible combinations?
    Spoiler: NP-Hard!
- Also, the difficulties inherent in dealing with the instance selection problem increase, as some techniques still treat such a selection based on empirical assumptions about the instances' locations. Moreover, some techniques treat such a selection as an isolated task, not fully incorporating its effects into the induced models (GARCIA *et al.*, 2012).

Introduction

The catch...

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

The Challenge...

- In this case, what means reducing complexity?
  Short Answer: Control the influence of samples during both training and prediction.
  - How to define a subset with K samples from the dataset with N samples ($N \geq K$)?
  - How to identify which specific combination of samples among the $\binom{N}{K}$ possible combinations?
    Spoiler: NP-Hard!
- Also, the difficulties inherent in dealing with the instance selection problem increase, as some techniques still treat such a selection based on empirical assumptions about the instances' locations. Moreover, some techniques treat such a selection as an isolated task, not fully incorporating its effects into the induced models (GARCIA *et al.*, 2012).
- Can we balance both the complexity of the induced model and the ability to generalize?

Introduction

The catch...

Despite the remarkable performance achieved by instance-based models, they all suffer from the scalability problem since they build hypotheses directly from the training instances themselves, thus implying that the hypotheses' complexity can grow with the data (NORVIG; RUSSELL, 2017).

The Challenge...

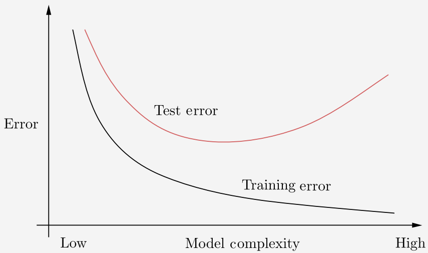
- In this case, what means reducing complexity?
  Short Answer: Control the influence of samples during both training and prediction.
  - How to define a subset with K samples from the dataset with N samples ($N \geq K$)?
  - How to identify which specific combination of samples among the $\binom{N}{K}$ possible combinations?
    Spoiler: NP-Hard!
- Also, the difficulties inherent in dealing with the instance selection problem increase, as some techniques still treat such a selection based on empirical assumptions about the instances' locations. Moreover, some techniques treat such a selection as an isolated task, not fully incorporating its effects into the induced models (GARCIA *et al.*, 2012).
- Can we balance both the complexity of the induced model and the ability to generalize?
  Hopefully, we can!

Controlling complexity

Classical expected result

The training error tends to zero as the model complexity increases; for complex enough models with a large number of free parameters, a perfect fit to the training data is possible, see **Figure 1**.

Figure 1: Model complexity vs. Error.



Adapted from Theodoridis (2020).

"Occam's razor" principle (MACKAY, 1992)

The Occam's razor principle states that unnecessarily complex models should not be preferred to simpler ones. Thus, it encourages us to employ mechanisms to penalize over-complex models as a way out to benefit model generalization:

$$h^{\star} = \underset{h \, \in \, \mathcal{H}}{\arg\min} \;\; \mathrm{loss}(h) + \lambda \; \mathrm{complexity}(h), \qquad (1)$$

where $\lambda \in \mathbb{R}^{+}$ serves as a conversion rate between loss and hypothesis complexity.

Motivation

Conducting wire. . .

- Instance-based learners habitually adopt instance selection techniques to reduce complexity and avoiding overfitting;
- Their most recent and well-known formulations seek to obtain a low-rank linear system by selecting samples to impose some sparsity in training and prediction alongside regularization;
- This thesis revisits the instance selection and regularization strategies to extend two instance-based models, namely, Least-Squares Support Vector Machines and Minimal Learning Machine;
- Firstly, we developed an instance selection algorithm, that later it will derive instance-based models;
- After, we revist the Least-Squares Support Vector Machines and we investigate the effects of such an instance selection in the model performance;
- Finally, we revisit the Lightweight Minimal Learning Machine model and investigates regularization usage by the pattern tasks without discarding data.

Objectives

Main goal

Having that said, the motivation for this research is to derive an instance-based models that can embody an instance selection mechanism while controlling the model's complexity.

Specific ones

- Propose a new instance selection method;
- Extend an instance-based model by employing an instance selection;
- Integrate an instance selection mechanism during the learning phase as regularization;
- Evaluate the folowing aspects of each proposal: the prediction error performance, the goodness-of-fit of estimated vs. measured values, and the norm values which are related to the sparsity (model complexity).

Reducing Complexity By Instance Selection

[Proposal#1] CCIS: Selecting class-corner samples

The Class-Corner Instance Selection

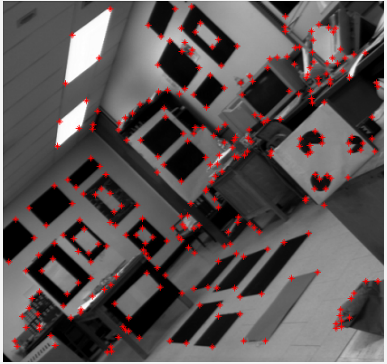
Formulation

The Class-Corner Instance Selection (CCIS) (OLIVEIRA *et al.*, 2018) is mainly based on an image corner detector FAST (ROSTEN; DRUMMOND, 2006). The main idea is to use the definition of what is a corner in FAST and then apply the same reasoning as the Instance Selection algorithm.

Attention (!)

However, it turns out that FAST formulation only deals with image data, i.e., two dimensional samples that are uniformly spaced in a grid. To overcome such limitations, we extended FAST so that we can apply it to high-dimensional inputs in a straightforward way.

Figure 2: Corner detection in a gray scale image. Corners are highlighted in red asterisks ($*$).



Taken from Kahaki *et al.* (2014).

The Class-Corner Instance Selection

Formulation

The Class-Corner Instance Selection selects a subset $\mathcal{PS}$ from a given set $\mathcal{D}$, i.e., based on a function $\Gamma(\cdot)$ that identifies the corners candidates and actual corner samples. Formally, let $\mathcal{PS}$ be defined as:

$$\mathcal{PS} = \Big\{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D} \;\Big|\; \Gamma(\mathbf{x}_i) > P\Big\}. \tag{2}$$

such that

$$\Gamma(\mathbf{x}) = \sum_{\mathbf{x}_i} \mathbb{1}[\mathbf{y} \neq \mathbf{y}_i],\;\; \mathbf{x}_i \;\in \mathcal{N}_R(\mathbf{x}), \tag{3}$$

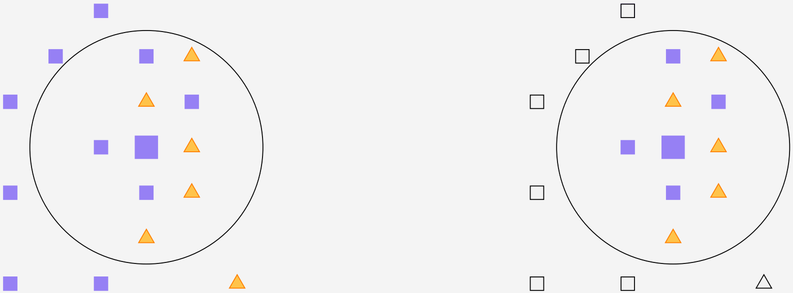
where $\mathbb{1}[\cdot]$ is the indicator function that it is equal to 1 if its argument is true or 0 otherwise and $\Gamma(\cdot)$ accounts the number of neighbors with different class labels than the query sample $\mathbf{x}$ inside the R-ball defined by $\mathcal{N}_R(\cdot)$:

$$\mathcal{N}_R(\mathbf{x}) = \Big\{\mathbf{x}_i \in \mathcal{NN}_K(\mathbf{x}) \;\Big|\; 0 < \|\mathbf{x}_i - \mathbf{x}\|_2 \leq R\Big\}, \tag{4}$$

where $R \in \mathbb{R}_+$ is the radius of the circle mask and $\mathcal{NN}_K(\cdot)$ yields the set with K nearest neighbors. The authors discuss in Oliveira *et al.* (2018) that by adopting $P = 0$ and default values for K and P to derive feasible subsets that share this class-corner feature.

CCIS in Action!

Figure 3: CCIS for some artificial data.



(a) Finding the K nearest neighbors and the radius defined by R.

(b) The samples outside the radius defined by R are not taken into account.

Source: Own authorship.

[Proposal#2] CC-LSSVM: When LSSVM meets CCIS

The Least-Squares Support Vector Machine

Foundation

- The LSSVM primal problem is defined as

$$\min_{\mathbf{w},\boldsymbol{\xi}} \mathcal{J}(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^{\intercal}\mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\xi_i^2$$
$$\text{s.t. } \mathbf{w}^{\intercal}\phi(\mathbf{x}_i) + b = y_i - \xi_i, i = 1, \ldots, N \tag{5}$$

whose solution (**in dual form**) is the saddle point of the following Lagrangian function:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \mathcal{J}(\mathbf{w},\boldsymbol{\xi}) + \sum_{i=1}^{N}\alpha_i\left(y_i - \mathbf{w}^{\intercal}\phi(\mathbf{x}_i) - b - \xi_i\right), \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$ are the Lagrangian multipliers. After eliminating $\mathbf{w}$ and $\boldsymbol{\xi}$, one obtains the Karush-Kuhn-Tucker (KKT) system $\mathbf{Au} = \mathbf{v}$ from the conditions for optimality:

$$\underbrace{\left[\begin{array}{c|c} 0 & \mathbf{1}^{\intercal} \\ \hline \mathbf{1} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{array}\right]}_{\mathbf{A}} \underbrace{\left[\begin{array}{c} b \\ \boldsymbol{\alpha} \end{array}\right]}_{\mathbf{u}} = \underbrace{\left[\begin{array}{c} 0 \\ \mathbf{Y} \end{array}\right]}_{\mathbf{v}}, \tag{7}$$

whose unique dense solution can be obtained as follows $\mathbf{u} = \mathbf{A}^{-1}\mathbf{v}$.

- Predicting new data mainly refers to employ the resulting $\boldsymbol{\alpha}$ and b:

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i\mathcal{K}(\mathbf{x},\mathbf{x}_i) + b\right). \tag{8}$$

Controlling Complexity in Least-Squares Support Vector Machines

Starting point

With respect to the LSSVM model, one can simply highlight that the model complexity term is with respect to the $\mathbf{w}$ in the *primal form* and with respect to the Lagrangian multipliers $\boldsymbol{\alpha}$ in the *dual form* since $\mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)$.

Controlling Complexity in Least-Squares Support Vector Machines

Starting point

With respect to the LSSVM model, one can simply highlight that the model complexity term is with respect to the $\mathbf{w}$ in the *primal form* and with respect to the Lagrangian multipliers $\boldsymbol{\alpha}$ in the *dual form* since $\mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)$.

Disclaimer

According to Mall e Suykens (2015) they can be separated into two groups: (*i*) **reduction methods** and (*ii*) **direct methods**.

- The reduction methods focus on training an usual LSSVM and, then, apply some pruning strategy, identifying the new support vectors so that a newly trained model can be derived;
- The direct method paradigm enforces on the sparsity from the beginning.

CCLSSVM

Motivation

In some literature, it is supported that pruning and reduced-set strategies for LSSVM work quite well (CARVALHO; BRAGA, 2009; YANG *et al.*, 2014). **The only limitation is that if an input sample is a SV (column), its constraint (row) must be in the system since losing that link degrades the final solution**.

CC-LSSVM: LSSVM meets Class-corner Instance Selection

- To keep that aforementioned link, we select the SVs and the restrictions by setting $\mathcal{SV} = \mathrm{CCIS}(\mathcal{D}, 0)$ and $\mathcal{PS} = \mathrm{CCIS}(\mathcal{D}, P)$, where P is a threshold so that $\mathcal{SV} \subset \mathcal{PS} \subset \mathcal{D}$.
- Then, we formulate following linear system $\boldsymbol{\Lambda}\boldsymbol{\omega} = \boldsymbol{\upsilon}$ so that,

$$\underbrace{\left[\begin{array}{c|c} 0 & \mathbf{1}^\intercal \\ \hline \mathbf{1} & \boldsymbol{\Psi} \end{array}\right]}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} b^\star \\ \boldsymbol{\alpha}^\star \end{bmatrix}}_{\boldsymbol{\omega}} = \underbrace{\begin{bmatrix} 0 \\ \mathbf{Y}_{\mathrm{SV}} \end{bmatrix}}_{\boldsymbol{\upsilon}}, \tag{9}$$

where $\mathbf{Y}_{\mathrm{SV}} = [y_1, y_2, \ldots, y_M]^\intercal$ are the labels from $\mathcal{SV}$ and

$$\Psi_{i,j} = \begin{cases} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i \neq \mathbf{x}_j; \\ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) + \gamma^{-1}, & \text{otherwise.} \end{cases}$$

with $\mathbf{x}_i \in \mathcal{PS}$ and $\mathbf{x}_j \in \mathcal{SV}$ and $\gamma \in \mathbb{R}_+$ is the same cost parameter in (LS)SVM.

- The solution is obtained by the usual least-squares:

$$\hat{\boldsymbol{\omega}} = (\boldsymbol{\Lambda}^\intercal\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\intercal\boldsymbol{\upsilon}. \tag{10}$$

Out-of-sample prediction and computational complexity of CC-LSSVM

Out-of-sample prediction

In possession of the estimated dual variables $\boldsymbol{\alpha}^\star$, bias $b^\star$, and the SVs from $\mathcal{SV}$, one can build the final predictive model as

$$f(\mathbf{x}) = \mathrm{sign}\left(\sum_{j=1}^{M} \alpha_j^\star \mathcal{K}(\mathbf{x}, \mathbf{x}_j) + b^\star\right), \mathbf{x}_j \in \mathcal{SV}. \tag{11}$$

Computational complexity

As CC-LSSVM consists of two steps: (*i*) the class-corner support vector selection and (*ii*) computing the solution of a less constrained linear system, the overall complexity is given by the sum of the complexity of these two steps. However, neither a sparse solution neither a less constraint model is always possible according to the problem's nature. To simplify our notation, we adopt M as the number of selected SVs for further analysis. **Later, we discuss how M behaves in real-world experiments and how the worst-case affects the overall model performance, especially via parameter tuning. From the carried out experiments, we conclude that the overall complexity of CC-LSSVM runs in $\mathcal{O}(M^3)$ since** $\mathcal{O}(N\log(N)) + \mathcal{O}(M^3) \sim \mathcal{O}(M^3)$.

Experiments for CC-LSSVM

We carried out three types of experiments to evaluate different and to investigate how CC-LSSVM behaves concerning the some aspects. Each of the following experiments addresses such concerns. Moreover, we performed all experiments using a Macbook Pro with an Intel Core i5 2.4 GHz, 8 GB of RAM, and running macOS Sierra 10.12.6 with MATLAB Version 8.3.0.

Experiment Setup

- **The typical black-box assessment on some datasets**: Here, we are interested in assessing CC-LSSVM against some state-of-the-art dual LSSVM proposals through accuracy and sparseness in a typical black-box test fashion on some "toy-size" and large-size datasets.
- **Empirical decision boundary quality assessment**: This experiment investigates visually the quality of solutions produced by the some state-of-the-art dual LSSVM proposals and CC-LSSVM by empirically evaluating the decision boundaries and, consequently, the support vector selection.
- **Hyperparameter sensitivity**: In this experiment, we intend to identify the influence of hyperparameter tuning concerning the resulting CC-LSSVM model.

Typical black-box assessment

Table 1: Description of the datasets: name, acronym, input dimensionality and number of training and test samples.

DATASET	ACRONYM	#Dim	# Tr	#Te
Banana	BAN	2	4239	1061
Breast Cancer Winscousin	BCW	9	550	138
German	GER	24	800	200
Haberman's Survival	HAB	3	244	62
Heart	HEA	13	216	54
Hepatitis	HEP	19	65	15
Ionosphere	ION	34	281	70
Pima Indians Diabetes	PID	9	614	154
Ripley	RIP	2	999	250
Two Moon	TMN	2	800	201
Vertebral Column	VCP	6	248	62

Source: Oliveira *et al.* (2018).

LSSVM variants used in the typical black-box assessment

Table 2: LSSVM variants and their hyperparameter configurations.

MODEL	DESCRIPTION	PARAMETER VALUE
CC-LSSVM	Number of neighbors.	$K = 16$.
	Variability threshold.	$P = 9$.
	Distance threshold.	R obtained by (OLIVEIRA *et al.*, 2018).
FSA-LSSVM	ϵ-insensitive criterion.	$\epsilon = 0.5$.
SSD-LSSVM	Subset size.	$\lfloor 4 \times \sqrt{N} \rfloor$ (i.e., $k = 4$).
CCP-LSSVM	Sampling ratio.	$M = 0.3N$.
	Sparse approximation.	Orthogonal Matching Pursuit.
	Compressive sampling.	Random gaussian matrix.

Source: Oliveira *et al.* (2018).

Toy-size datasets

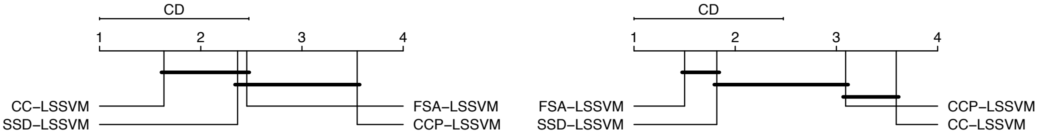
Table 3: Performance on some *toy-size* datasets. For each dataset, the best performing models are in boldface. We recall that all values are the average for 30 independent realizations.

DATASET	CC-LSSVM		FSA-LSSVM		SSD-LSSVM		CCP-LSSVM	
	ACC	SPR	ACC	SPR	ACC	SPR	ACC	SPR
BAN	**0.89 ± 0.01**	0.93 ± 0.00	**0.89 ± 0.01**	0.93 ± 0.00	0.90 ± 0.01	**0.96 ± 0.00**	0.68 ± 0.17	0.72 ± 0.00
BCW	**0.96 ± 0.01**	0.75 ± 0.01	0.95 ± 0.02	**0.96 ± 0.01**	0.94 ± 0.02	0.84 ± 0.00	0.81 ± 0.06	0.71 ± 0.00
GER	**0.75 ± 0.02**	0.00 ± 0.00	0.72 ± 0.02	0.81 ± 0.01	0.70 ± 0.00	**0.86 ± 0.00**	0.60 ± 0.07	0.71 ± 0.00
HAB	0.73 ± 0.03	0.31 ± 0.02	**0.75 ± 0.03**	0.79 ± 0.02	0.74 ± 0.03	**0.80 ± 0.01**	0.58 ± 0.16	0.71 ± 0.00
HEA	**0.84 ± 0.05**	0.00 ± 0.00	0.56 ± 0.00	**0.87 ± 0.02**	0.57 ± 0.02	0.73 ± 0.00	0.71 ± 0.08	0.71 ± 0.00
HEP	**0.70 ± 0.10**	0.00 ± 0.00	0.60 ± 0.00	0.65 ± 0.04	0.60 ± 0.00	0.55 ± 0.02	0.53 ± 0.12	**0.72 ± 0.00**
ION	**0.93 ± 0.04**	0.40 ± 0.01	0.68 ± 0.03	**0.85 ± 0.01**	0.69 ± 0.10	0.77 ± 0.00	0.73 ± 0.06	0.71 ± 0.00
PID	**0.76 ± 0.02**	0.24 ± 0.01	0.68 ± 0.02	0.83 ± 0.01	0.65 ± 0.00	**0.84 ± 0.00**	0.66 ± 0.04	0.71 ± 0.00
RIP	0.90 ± 0.02	0.70 ± 0.01	**0.91 ± 0.02**	**0.92 ± 0.01**	**0.91 ± 0.01**	0.90 ± 0.00	0.75 ± 0.16	0.72 ± 0.00
TMN	0.99 ± 0.01	0.97 ± 0.00	0.99 ± 0.01	**0.98 ± 0.01**	**1.00 ± 0.00**	0.89 ± 0.00	0.98 ± 0.04	0.72 ± 0.00
VCP	**0.87 ± 0.03**	0.59 ± 0.01	0.84 ± 0.03	**0.92 ± 0.01**	0.85 ± 0.03	0.78 ± 0.01	0.59 ± 0.16	0.71 ± 0.00
AVG RNK.	**1.64**	**3.59**	**2.45**	**1.50**	**2.36**	**1.82**	**3.55**	**3.09**

Source: Oliveira *et al.* (2018).

Toy-size datasets

Figure 4: Critical difference plots with respect to the accuracy rankings (a) and the sparsity rankings (b) from Table 3. We recall that those variants which are not joined by a bold line can be regarded as different.



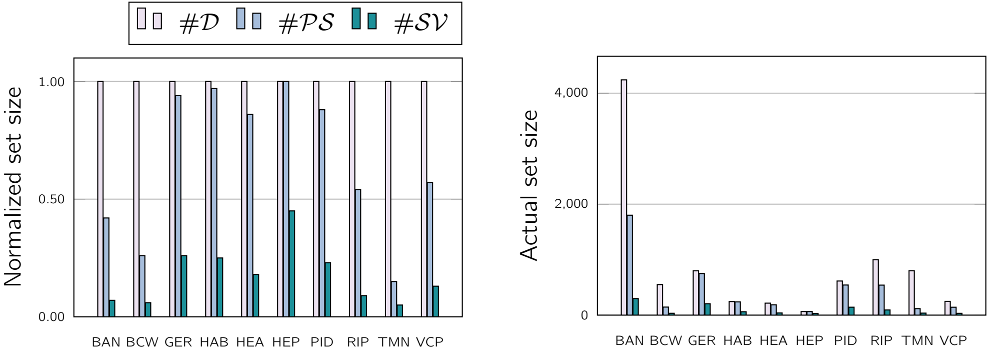
(a) Accuracy rankings.

(b) Sparsity rankings.

Source: Oliveira *et al.* (2018).

Set-size reduction by CCIS in CC-LSSVM.

Figure 6: Bar plots showing the original training set $\mathcal{D}$, the number of elements in $\mathcal{PS}$, and the number of support vectors in $\mathcal{SV}$ for CC-LSSVM. We show both the scaled and the actual sizes for each *toy size* dataset.



(a) Normalized set sizes for $\mathcal{D}$, $\mathcal{PS}$, and $\mathcal{SV}$.

(b) Normalized set sizes for $\mathcal{D}$, $\mathcal{PS}$, and $\mathcal{SV}$.

Source: Adapted from Oliveira *et al.* (2018).

Large-size datasets

Table 4: Large-size dataset description: input/output dimensionality and number of training/test samples and required sparsity level.

DATASET	#Dim	# Tr	#Te	$\#\mathcal{SV}$
ADULT	123	32561	16281	300 (0.92%)
IJCNN	22	49990	91790	400 (0.80%)
SHUTTLE	9	43500	14500	200 (0.46%)
USPS8	256	7291	2007	200 (2.07%)
VEHICLE	100	78823	19705	400 (0.51%)

Source: Oliveira *et al.* (2018).

Dealing with large-size datasets in CC-LSSVM

We perform a greedy class-corner SV selection in a stratification fashion (with J disjoint subsets, i.e., $\mathcal{D} = \bigcap_{i=1}^{J} \mathcal{S}_i$). To provide a fixed-size approach, we sort descendingly based on $\Gamma(\cdot)$ to rescue the first M samples.

Large-size datasets

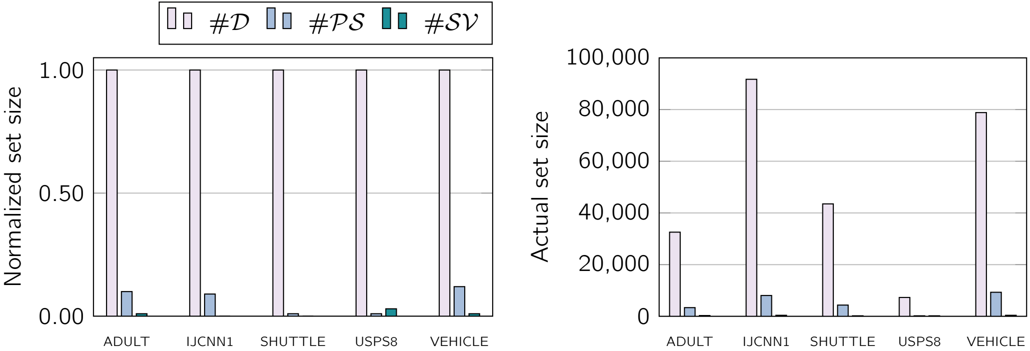
Table 5: Performance on some large datasets. For each dataset, the best performing models are in boldface. We recall that all values are the average for 20 independent realizations.

DATASET	FCC-LSSVM	FSA-LSSVM	SSD-LSSVM
ADULT	**0.76** ± **0.00**	0.65 ± 0.04	**0.76** ± **0.00**
IJCNN	0.89 ± 0.00	0.78 ± 0.07	**0.93** ± **0.00**
SHUTTLE	0.99 ± 0.00	**1.00** ± **0.00**	**1.00** ± **0.00**
USPS8	0.93 ± 0.00	**0.99** ± **0.00**	0.93 ± 0.00
VEHICLE	**0.84** ± **0.00**	0.62 ± 0.04	0.54 ± 0.00

Source: Oliveira *et al.* (2018).

Set-size reduction by CCIS in CC-LSSVM for large-size datasets

Figure 8: Bar plots showing the set sizes for FCC-LSSVM: the training set $\mathcal{D}$, the prototype vectors' set $\mathcal{PS}$, and the support vectors' set $\mathcal{SV}$. We show both the scaled and the actual sizes for each large-size dataset.



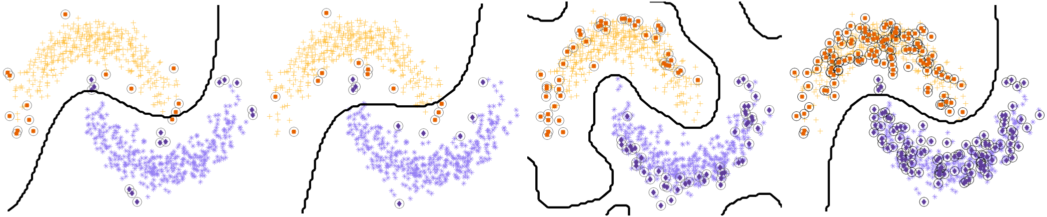
(a) Normalized set sizes for $\mathcal{D}$, $\mathcal{PS}$, and $\mathcal{SV}$.

(b) Actual set sizes for $\mathcal{D}$, $\mathcal{PS}$, and $\mathcal{SV}$.

Source: Oliveira *et al.* (2018).

Empirical decision boundary quality assessment I

Figure 10: Two moon dataset and the produced decision boundaries by some LSSVM variants.

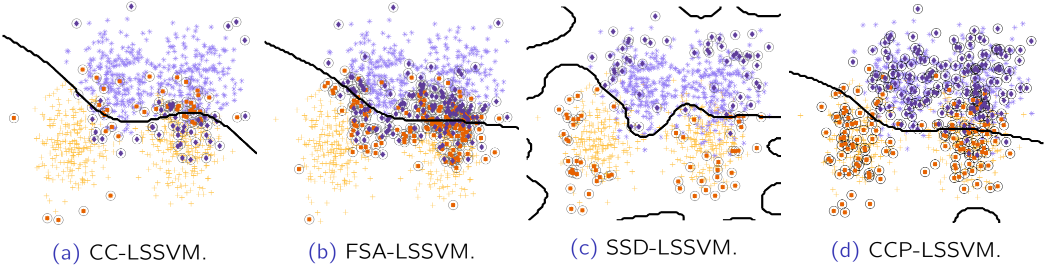


(a) CC-LSSVM. (b) FSA-LSSVM. (c) SSD-LSSVM. (d) CCP-LSSVM.

Source: Oliveira *et al.* (2018).

Empirical decision boundary quality assessment II

Figure 12: Ripley dataset and the produced decision boundaries by some LSSVM variants.



(a) CC-LSSVM.

(b) FSA-LSSVM.

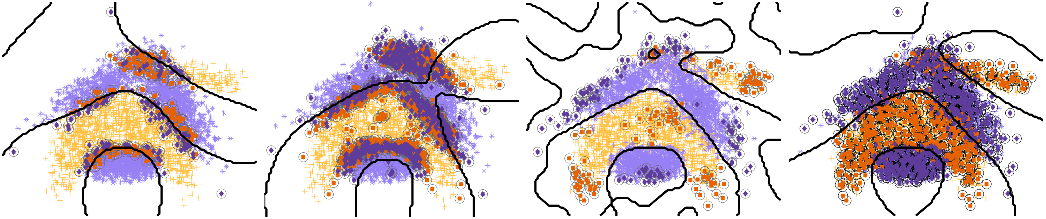
(c) SSD-LSSVM.

(d) CCP-LSSVM.

Source: Oliveira *et al.* (2018).

Empirical decision boundary quality assessment III

Figure 14: Banana dataset and the produced decision boundaries by some LSSVM variants.



(a) CC-LSSVM.

(b) FSA-LSSVM.

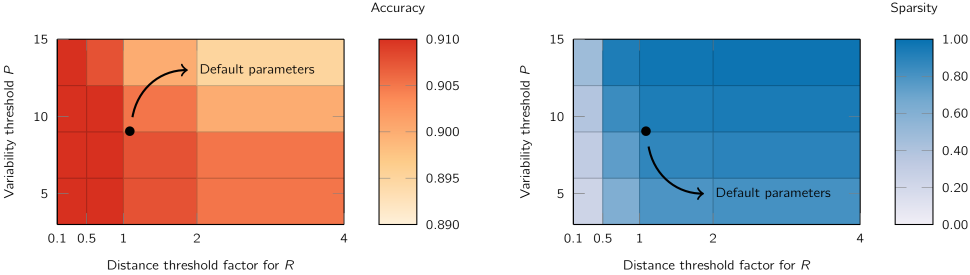
(c) SSD-LSSVM.

(d) CCP-LSSVM.

Source: Oliveira *et al.* (2018).

The hyperparameter influence for CC-LSSVM

Figure 16: Hyperparameter influence concerning both Accuracy and Sparsity in CC-LSSVM using Ripley dataset. The black point stands for the default values empirically determined. The variability threshold (y-axis) is P, while the distance threshold (x-axis) is a factor of R.



(a) Accuracy experiment.

(b) Sparsity experiment.

Source: Adapted from Oliveira *et al.* (2018).

Computational complexity analysis

Table 6: LSSVM variant comparison in terms of memory requirement, training cost, and prediction cost. We recall that N stands for the cardinality of the training set, while M is the number of support vectors.

MODEL	MEMORY	TRAINING	PREDICTION
LSSVM (DUAL)	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
FSA-LSSVM (JIAO *et al.*, 2007)	$\mathcal{O}(M^2)$	$\mathcal{O}(M^3)$	$\mathcal{O}(M)$
CCP-LSSVM (YANG *et al.*, 2014)	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(M)$
SSD-LSSVM (MALL; SUYKENS, 2015)	$\mathcal{O}(MN)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M)$
CC-LSSVM	$\mathcal{O}(MN)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M)$
FSOCC-LSSVM	$\mathcal{O}(MN)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M)$

Source: Oliveira *et al.* (2018).

Reducing Complexity by Instance Regularization

The Minimal Learning Machine

Foundation

- MLM estimates $h(\cdot)$ for the target function $f(\cdot)$, from the data $\mathcal{D}$ through the distance domain;
- By employing pairwise distance matrices of each point of $\mathcal{D}$, namely, $\mathbf{D}$ and $\mathbf{\Delta}$, both representing the Euclidean distance – in the notation of $\mathrm{d}(\cdot,\cdot)$ – of each point from $\mathcal{D}$ to the i-th reference point of $\mathcal{D}$.
- Such that, $D_{i,j} = \mathrm{d}(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{\Delta}_{i,j} = \mathrm{d}(\mathbf{y}_i, \mathbf{y}_j)$, then they have $N \times N$ dimensions;
- Furthermore, it assumes that the mapping between the distance matrices has a linear structure for each response, the MLM model can be rewritten in the form:

  $$\mathbf{\Delta} = \mathbf{D}\mathbf{B} + \mathbf{E}, \tag{12}$$

  where $\mathbf{E}$ is a residuals matrix.
- **We name this version Full-MLM in this thesis for differentiation purposes**.

The Minimal Learning Machine

MLM learning algorithm

Since we assume such a linear mapping between domains, we rewrite the regression model as follows:

$$\min_{\mathbf{B}} \ \mathcal{J}(\mathbf{B}) = ||\mathbf{D}\mathbf{B} - \mathbf{\Delta}||^2_{\mathcal{F}}, \tag{13}$$

$\mathbf{B}$ can be estimated by: $\hat{\mathbf{B}} = \mathbf{D}^{-1}\mathbf{\Delta}$.

MLM out-of-sample prediction

Predicting the outputs for new input data mainly refers to project the new data point through the mapping and estimate the image of such a projection[a] by minimizing the objective function below:

$$\hat{\mathbf{y}} = h(\mathbf{x}) = \arg\min_{\mathbf{y}} || \,\Psi(\mathbf{y}) - \Phi(\mathbf{x})\,\mathbf{B}\, ||_2, \tag{14}$$

where both $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^N$ and $\Psi : \mathbb{R}^S \rightarrow \mathbb{R}^N$ are distance mapping functions such that $\Phi(\mathbf{x}) = \left[\mathrm{d}(\mathbf{x}, \mathbf{x}_1), \mathrm{d}(\mathbf{x}, \mathbf{x}_2), \ldots, \mathrm{d}(\mathbf{x}, \mathbf{x}_N)\right]^{\mathsf{T}}$ while $\Psi(\mathbf{y}) = \left[\mathrm{d}(\mathbf{y}, \mathbf{y}_1), \mathrm{d}(\mathbf{y}, \mathbf{y}_2), \ldots, \mathrm{d}(\mathbf{y}, \mathbf{y}_N)\right]^{\mathsf{T}}$.

[a] This problem can be treated as a multilateration (NIEWIADOMSKA-SZYNKIEWICZ; MARKS, 2009).

Controlling Complexity in Minimal Learning Machines

Starting point

Concerning the MLM model, as a linear model, the complexity term is related to **B**. Thus, imposing sparsity in **B** is also a manner to reduce the complexity (especially when there are zero coefficients).

Controlling Complexity in Minimal Learning Machines

Starting point

Concerning the MLM model, as a linear model, the complexity term is related to **B**. Thus, imposing sparsity in **B** is also a manner to reduce the complexity (especially when there are zero coefficients).

Disclaimer

Even though some variants employ the reference point selection (data discard) as the principal aspect of the MLM learning algorithm, they pretty much embody a lower rank linear system through such a selection as a manner to also impose some sparsity in both training and speed-up prediction. Increasing sparsity is a manner to reduce complexity.

Controlling Complexity in Minimal Learning Machines

Starting point

Concerning the MLM model, as a linear model, the complexity term is related to **B**. Thus, imposing sparsity in **B** is also a manner to reduce the complexity (especially when there are zero coefficients).

Disclaimer

Even though some variants employ the reference point selection (data discard) as the principal aspect of the MLM learning algorithm, they pretty much embody a lower rank linear system through such a selection as a manner to also impose some sparsity in both training and speed-up prediction. Increasing sparsity is a manner to reduce complexity.

Side effect

However, employing both reduced dataset and regularization in MLM is likely to lead to underfitting since if **B** acts as a poor transformation, i.e., **B** can not correctly map both input and output spaces, the prediction is strongly impaired. Also, note that by employing both strategies as they are presented, they simply restrict the hypothesis that the learning phase can find.

Controlling Complexity in Minimal Learning Machines – What do we have so far?

Random MLM (JUNIOR *et al.*, 2015)

The model itself relies on the approximate solution provided by the ordinary least-squares estimate of **B** because both **D** and **Δ** have dimensions $N \times K$ (K corresponds to a hyperparameter), resulting in:

$$\min_{\mathbf{B}} \; \mathcal{J}_{\text{Rand}}(\mathbf{B}) = ||\mathbf{D}\mathbf{B} - \mathbf{\Delta}||_{\mathcal{F}}^{2}, \tag{15}$$

which yields the following solution:

$$\hat{\mathbf{B}}_{\text{Rand}} = (\mathbf{D}^{\intercal}\mathbf{D})^{-1}\mathbf{D}^{\intercal}\mathbf{\Delta}. \tag{16}$$

Rank-MLM (ALENCAR *et al.*, 2015)

Its main prominent feature is a training procedure which contains a regularized cost as follows:

$$\min_{\mathbf{B}} \; \mathcal{J}_{\text{Rank}}(\mathbf{B}, \lambda) = ||\mathbf{D}\mathbf{B} - \mathbf{\Delta}||_{\mathcal{F}}^{2} + \lambda||\mathbf{B}||_{\mathcal{F}}^{2} \tag{17}$$

which yields the following solution where **I** is a $N \times N$ identity matrix:

$$\hat{\mathbf{B}}_{\text{Rank}} = (\mathbf{D}^{\intercal}\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^{\intercal}\mathbf{\Delta}. \tag{18}$$

Controlling Complexity in Minimal Learning Machines – What do we have so far?

Weighted MLM (*w*MLM) (GOMES *et al.*, 2015)

It adopts the generalized least-squares fit between the input and output spaces as follows:

$$\min_{\mathbf{B}} \ \mathcal{J}_{\text{Weighted}}(\mathbf{B}, \mathbf{W}) = ||\mathbf{W}(\mathbf{DB} - \boldsymbol{\Delta})||_{\mathcal{F}}^{2} \tag{19}$$

where **W** is a symmetric positive definite diagonal matrix with each element $W_{i,i}$ representing the weight of each training sample $\mathbf{x}_i$. The above formulation yields the following solution:

$$\hat{\mathbf{B}}_{\text{Weighted}} = (\mathbf{D}^{\intercal}\mathbf{W}\mathbf{D})^{-1}\mathbf{D}^{\intercal}\mathbf{W}^{\intercal}\boldsymbol{\Delta}. \tag{20}$$

$\ell_{1/2}$-MLM (DIAS *et al.*, 2018)

Deriving out of pruning techniques, the $\ell_{1/2}$-MLM seeks the following optimization problem solution:

$$\min_{\mathbf{B}} \ \mathcal{J}_{\ell_{1/2}}(\mathbf{B}, \lambda, \alpha, K) = ||\mathbf{DB} - \boldsymbol{\Delta}||_{\mathcal{F}}^{2} + \lambda||\mathbf{B}||_{1/2}^{1/2} \tag{21}$$

where $\lambda \in \mathbb{R}^{+}$ is where is a trade-off parameter, α and K both controls the prunning processing. Since there is no closed-form to solve (21), one must apply the gradient descent algorithm using α and K.

[Proposal#3] LW-MLM: When MLM meets regularization by sample

Controlling Complexity in Minimal Learning Machines – What do we have so far?

Proposal #3: Lightweight MLM (LW-MLM) (FLORENCIO V *et al.*, 2020)

- Provide a new MLM formulation able to employ regularization on samples;
- Good to integrate an instance selection mechanism during the learning phase as regularization;
- The learning algorithm in LW-MLM simply envoles estimating a **B** from a given **P** as follows:

$$\min_{\mathbf{B}} \ \mathcal{J}_{\mathrm{LW}}(\mathbf{B}, \mathbf{P}) = ||\mathbf{D}\mathbf{B} - \boldsymbol{\Delta}||_{\mathcal{F}}^2 + ||\mathbf{P}\mathbf{B}||_{\mathcal{F}}^2, \tag{22}$$

which is achieved by the following solution:

$$\hat{\mathbf{B}} = (\mathbf{D}^\intercal\mathbf{D} + \mathbf{P}^\intercal\mathbf{P})^{-1}\mathbf{D}^\intercal\boldsymbol{\Delta}. \tag{23}$$

where **P** can be derived by a vector $\mathbf{p} \in \mathbb{R}^N$ i.e., $\mathbf{P} = \mathrm{diag}(\mathbf{p})$.

On the selection of P

Experiment Setup

- **By the normal random-based one** Related to the compressive sensing described in Yang *et al.* (2014), we seek to obtain the **P** by randomly selecting values the diagonal, letting the other indexes with 0. In this case, **p** is defined as $p_i \sim \mathcal{N}(0, 1)$.
- **By the nonlinear parts of** f Júnior (2014) presented an instance selection algorithm based on distance computations and non-parametric hypothesis testing. By analyzing the *p-values* from the non-parametric hypothesis test, the algorithm indicates which samples correspond to the most linear part of the target function $f(\cdot)$. Consequently, one can find the less linear ones. Therefore, to overcome such a situation, we adopted a transformation to address such a condition by penalizing the samples with *p-values* below a given linear threshold t, that is,

$$\mathcal{KS}^k_{t,\lambda}(\mathbf{x}, \mathbf{y}) = \begin{cases} \Pi(\mathbf{x}, \mathbf{y}) + \lambda, & \text{if } \Pi(\mathbf{x}, \mathbf{y}) < t, \\ \Pi(\mathbf{x}, \mathbf{y}), & \text{otherwise.} \end{cases} \tag{24}$$

Speeding up the out-of-sample prediction

Experiments

- The *lightweight* in LW-MLM is not just related to the regularized values in $\mathbf{B}$ (linear mapping coefficients) but also related to the speedup out-of-sample prediction. Since we employ all samples in the LW-MLM learning algorithm, we believe that LW-MLM learns the whole known geometric structure, i.e., the domain knowledge is “fully” represented in $\mathbf{B}$. With that in mind, we truly support that $\mathbf{B}$ by itself. Thus, in this setting, LW-MLM can successfully map the linear structure.
- Such an assumption encourages us to discard some components from the out-of-sample prediction procedure since most RP projections will result in zero error. First, let us define the discard function $\kappa : \mathbb{R}^N \rightarrow \mathbb{R}^K$ as
$$\kappa(\mathbf{a}) = (a_{i_1}, a_{i_2}, \ldots, a_{i_K})^\intercal, \tag{25}$$
where $i_1, i_2, \ldots, i_K, \ldots, i_N$ form a random permutation of $\{1, \ldots, N\}$. Now, the location of $\hat{\mathbf{y}}$ can be estimated by minimizing the following objective function:
$$\hat{\mathbf{y}} = h(\mathbf{x}) = \arg\min_{\mathbf{y}} \| \ \kappa\left(\Psi(\mathbf{y}) - \Phi(\mathbf{x})\,\mathbf{B}\right) \|_2 \ . \tag{26}$$
- We discuss such a speedup procedure in the experiments. We show the relationship when varying the number of components and the prediction error associated.

Assessing

Experiments

- **Typical black-box assessment**: Here, we assess LW-MLM against some variants of MLM, analyzing the prediction error and the goodness-of-fit of estimated vs. measured values in a typical black-box test fashion on some datasets;
- **Visual qualitative analysis**: In this experiment, we examine the importance of regularization by analyzing the level of complexity (via *smoothness*) of the estimated function while we identify the difference between LW-MLM and the other formulations empirically;
- **The relevance of RPs in the prediction step**: This experiment examines the influence of the number of RPs during out-of-sample prediction in the resulting LW-MLM model;
- **The prediction error in high dimensional feature**: Here, we conclude our experiments by analyzing how LW-MLM deals with high dimension datasets, i.e., $\mathbf{x} \in \mathbb{R}^D$ s.t. D from $\approx$ 100 up to 4000.

Typical black-box assessment

Table 7: Dataset descriptions for black-box assessment.

DATASET	ACRONYM	# DIM	# TR	# TE
Abalone	ABA	8	3000	1177
AutoMPG	MPG	7	350	42
Boston housing	BTH	13	400	106
concrete	CON	8	700	330
cpu_act	CPU	12	5000	3192
delta ailerons	DAI	5	5000	2129
delta elevators	DEL	6	6000	3517
kinematics	KIN	8	4500	3692
motor_UPDRS	MUP	20	3000	2875
puma8NH	8NH	8	4500	3692
stock	STO	9	600	350
total_UPDRS	TUP	20	3000	2875
winequality_red	WRE	11	1000	599
winequality_white	WWH	11	3500	1398

Source: Florencio V *et al.* (2020).

MLM variants and their hyperparameters configurations

Table 8: MLM variants and their hyperparameters configurations.

MODEL	DESCRIPTION	HYPERPARAMETER VALUE
Full-MLM	K: Number of RPs.	$K = N$.
Random-MLM	K: Number of RPs chosen randomly from a range through grid search and cross-validation.	$K \in \{\lfloor 0.05 \times i \times N \rfloor\}_{i=1}^{10}$
Rank-MLM	K: Number of RPs chosen randomly from a range through grid search and cross-validation C : Regularization parameter optimized by grid search and cross-validation.	$K \in \{\lfloor 0.05 \times i \times N \rfloor\}_{i=1}^{10}$ $C \in \{2^{-4}, 2^{-3}, \ldots, 2^{1}, 2^{2}\}$.
LW-MLM-1	**P**: Diagonal matrix with random values from a normal distribution with zero mean and unit variance.	$P_{i,j} = \begin{cases} \mathcal{N}(0,1), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}.$
LW-MLM-2	**P**: Diagonal matrix with values assigned via the KS-2-Sample-Test described in 24. k: Number of nearest points to compute distance.	$P_{i,j} = \begin{cases} \mathcal{KS}_0^k(\mathbf{x}_i, \mathbf{y}_i), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}.$ $k = \log_{10}(5 \times N)$.
LW-MLM-3	**P**: Diagonal matrix with values assigned via the KS-2-Sample-Test described in 24. k: Number of nearest points to compute distance. t: Linearity threshold.	$P_{i,j} = \begin{cases} \mathcal{KS}_t^k(\mathbf{x}_i, \mathbf{y}_i), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}.$ $k = \log_{10}(5 \times N)$. $t = \mathcal{OTSU}_k(\mathbf{X}, \mathbf{Y})$.

Source: Florencio V *et al.* (2020).

Result analysis

Assessment

- Regarding the statistical analysis of the results, we carried out the Friedman test to assess all performance metrics, namely, the RMSE, the R^2, and the adjusted norm of matrix **B** to better distinguish the comparison of the MLM models over 30 independent realizations. To do it so, we graphically represent such a comparison through the Critical Difference (CD) plot.
- Also, we highlight the norm values were scaled between 0 and 1 by subtracting 1 from division by the $\| \mathbf{B} \|_{\mathcal{F}}$ of Full-MLM (that is $\mathrm{NORM}(x) = 1 - x\| \mathbf{B} \|_{\mathcal{F}}{}^{-1}$) since it acts as an upper bound[a] because in other models either **D** and/or **Δ** are not squared matrices nor they present any regularization factor.

[a] $\| \mathbf{B} \|_{\mathcal{F}} = \| \mathbf{D}^{-1}\boldsymbol{\Delta} \|_{\mathcal{F}} \geq \| \mathbf{D}^{-1} \|_{\mathcal{F}} \| \boldsymbol{\Delta} \|_{\mathcal{F}}$.

Prediction error as performance measurement I

Table 9: Black-box experiment results for RMSE. We recall that the variants which are not joined by a bold line can be regarded as different in the CD plot bellow.

DATASETS	FULL-MLM	RANDOM-MLM	RANK-MLM	LW-MLM-1	LW-MLM-2	LW-MLM-3
ABA	2.24 ± 0.07	2.14 ± 0.06	2.12 ± 0.06	2.13 ± 0.06	2.22 ± 0.07	2.22 ± 0.07
MPG	2.64 ± 0.53	2.69 ± 0.50	2.62 ± 0.49	2.61 ± 0.47	2.57 ± 0.48	2.58 ± 0.48
BTH	3.11 ± 0.48	3.47 ± 0.60	3.57 ± 0.57	3.54 ± 0.52	3.31 ± 0.50	3.80 ± 0.49
CON	5.76 ± 0.53	6.40 ± 0.46	7.25 ± 0.38	7.21 ± 0.38	7.37 ± 0.41	7.94 ± 0.35
CPU	2.81 ± 0.06	2.86 ± 0.07	2.94 ± 0.07	2.91 ± 0.06	2.81 ± 0.06	2.81 ± 0.06
DAI	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*
DEL	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*	0.00 ± 0.0*
KIN	0.08 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.10 ± 0.00	0.11 ± 0.00
MUP	2.11 ± 0.06	2.34 ± 0.06	2.68 ± 0.05	2.65 ± 0.05	2.49 ± 0.06	2.52 ± 0.06
8NH	3.41 ± 0.04	3.36 ± 0.04	3.33 ± 0.04	3.33 ± 0.04	3.32 ± 0.03	3.37 ± 0.04
STO	0.66 ± 0.04	0.74 ± 0.04	0.83 ± 0.05	0.83 ± 0.05	0.79 ± 0.04	0.85 ± 0.05
TUP	2.85 ± 0.08	3.17 ± 0.07	3.68 ± 0.06	3.66 ± 0.07	3.48 ± 0.07	3.97 ± 0.07
WRE	0.61 ± 0.02	0.62 ± 0.02	0.62 ± 0.01	0.62 ± 0.02	0.61 ± 0.02	0.61 ± 0.02
WWH	0.61 ± 0.02	0.65 ± 0.01	0.67 ± 0.01	0.67 ± 0.01	0.61 ± 0.02	0.61 ± 0.02

* The values are not exactly zero but rather too small.

Source: Florencio V *et al.* (2020).

Prediction error as performance measurement II

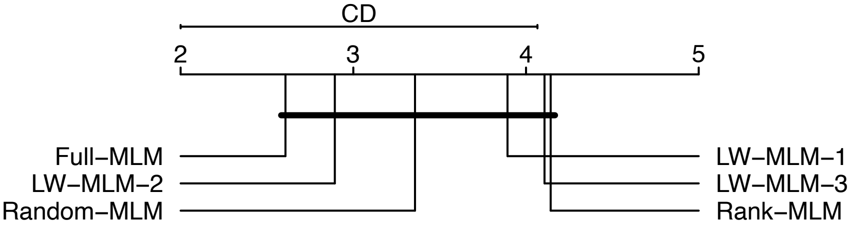


Figure 18: We recall that the variants which are not joined by a bold line can be regarded as different in the CD plot bellow.

Source: Florencio V *et al.* (2020).

Goodness-of-fit as performance measurement I

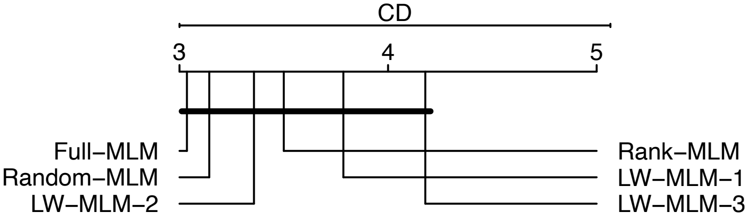
Table 10: Black-box experiment results for R^2.

DATASETS	FULL-MLM	RANDOM-MLM	RANK-MLM	LW-MLM-1	LW-MLM-2	LW-MLM-3
ABA	0.53 ± 0.02	0.57 ± 0.02	0.58 ± 0.02	0.57 ± 0.02	0.54 ± 0.02	0.54 ± 0.02
MPG	0.89 ± 0.05	0.88 ± 0.04	0.89 ± 0.04	0.89 ± 0.04	0.89 ± 0.04	0.89 ± 0.04
BTH	0.88 ± 0.04	0.85 ± 0.05	0.85 ± 0.05	0.85 ± 0.05	0.87 ± 0.04	0.83 ± 0.05
CON	0.88 ± 0.02	0.85 ± 0.02	0.81 ± 0.02	0.81 ± 0.02	0.81 ± 0.02	0.78 ± 0.02
CPU	0.98 ± 0.00	0.98 ± 0.00	0.97 ± 0.00	0.98 ± 0.00	0.98 ± 0.00	0.98 ± 0.00
DAI	0.68 ± 0.01	0.71 ± 0.01	0.71 ± 0.01	0.71 ± 0.01	0.69 ± 0.01	0.69 ± 0.01
DEL	0.59 ± 0.01	0.64 ± 0.01	0.64 ± 0.01	0.64 ± 0.01	0.59 ± 0.01	0.59 ± 0.01
KIN	0.90 ± 0.00	0.89 ± 0.00	0.88 ± 0.00	0.88 ± 0.00	0.87 ± 0.00	0.83 ± 0.01
MUP	0.93 ± 0.00	0.92 ± 0.00	0.89 ± 0.00	0.90 ± 0.00	0.91 ± 0.00	0.91 ± 0.01
8NH	0.63 ± 0.01	0.64 ± 0.01	0.65 ± 0.01	0.65 ± 0.01	0.65 ± 0.01	0.64 ± 0.01
STO	0.99 ± 0.00	0.99 ± 0.00	0.98 ± 0.00	0.98 ± 0.00	0.99 ± 0.00	0.98 ± 0.00
TUP	0.93 ± 0.00	0.91 ± 0.00	0.88 ± 0.01	0.89 ± 0.01	0.90 ± 0.00	0.87 ± 0.01
WRE	0.44 ± 0.03	0.40 ± 0.03	0.40 ± 0.03	0.41 ± 0.03	0.44 ± 0.03	0.44 ± 0.03
WWH	0.53 ± 0.02	0.47 ± 0.02	0.43 ± 0.02	0.43 ± 0.02	0.53 ± 0.02	0.53 ± 0.02

Source: Florencio V *et al.* (2020).

Goodness-of-fit as performance measurement II

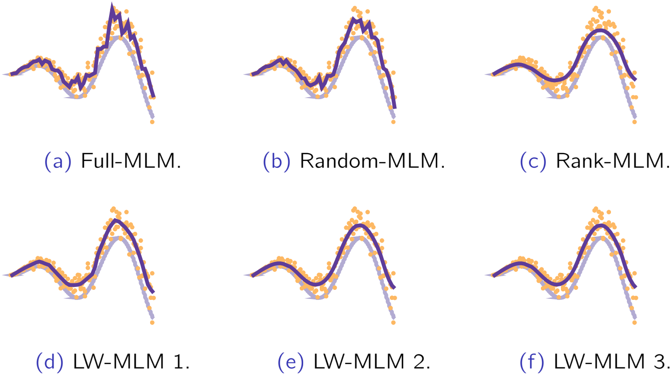
Figure 19: Again, we recall that the variants which are not joined by a bold line can be regarded as different in the CD plot bellow.



Source: Florencio V *et al.* (2020).

Visual qualitative analysis I

Figure 20: Artificial dataset I.



(a) Full-MLM.

(b) Random-MLM.

(c) Rank-MLM.

(d) LW-MLM 1.

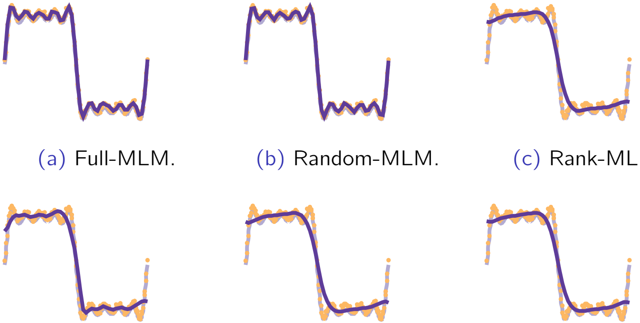
(e) LW-MLM 2.

(f) LW-MLM 3.

Source: Florencio V *et al.* (2020).

Visual qualitative analysis II

Figure 22: Artificial data set II.



(a) Full-MLM.

(b) Random-MLM.

(c) Rank-MLM.

(d) LW-MLM 1.

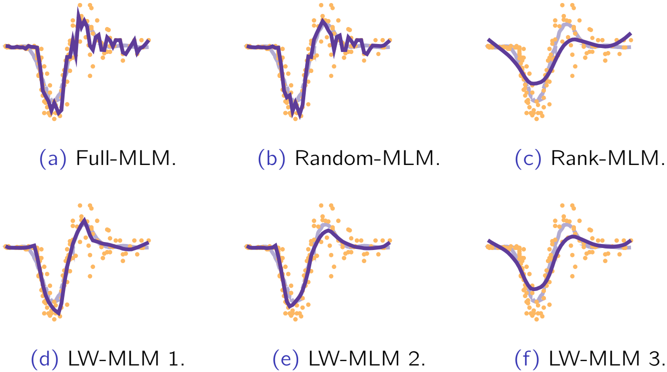
(e) LW-MLM 2.

(f) LW-MLM 3.

Source: Florencio V *et al.* (2020).

Visual qualitative analysis III

Figure 24: Mcycle dataset.



(a) Full-MLM.

(b) Random-MLM.

(c) Rank-MLM.

(d) LW-MLM 1.

(e) LW-MLM 2.

(f) LW-MLM 3.

Source: Florencio V *et al.* (2020).

Relevance of RPs during out-of-sample prediction I

The *lightweight* pillar

- We support that it learns the whole known geometric structure of the data itself, thus, encouraging us to discard some RPs in the out-of-sample prediction;
- In this experiment, we analyze how such a discard influences the error in LW-MLM;
- We vary the quantity of RPs from 2 points and, then, we increase it by a step of multiples of 5% of the actual dataset size to examine how the model behaves regarding the RMSE;
- **Finding:** **When adopting a full B, we can employ just 5% of the actual dataset size into the multilateration**.

High-dimension datasets I

Table 11: High dimension dataset descriptions.

DATASET	ACRONYM	# DIM	# TR	# TE
Residential (output: Sales price)	RS	105	297	75
Residential (output: Construction price)	RC	105	297	75
Communities Crime	CC	147	1772	443
Riboflavin	RB	4088	50	21

Source: Florencio V *et al.* (2020).

High-dimension datasets II

Table 12: Hyperparameters' space in the high dimension dataset experiment.

MODEL	DESCRIPTION	HYPERPARAMETER SPACE
RF	d: the maximum depth of the tree.	$d \in \{10, 30, 80, 100\}$.
	f: Max features.	$f \in \{2, 3\}$
	l: Minimum number of leaf node samples.	$l \in \{3, 4, 5\}$
	s: Number of minimum samples to split.	$s \in \{8, 10, 12\}$
	n: Number of trees in the forest.	$n \in \{100, 200, 300\}$
SVR	C: Regularization parameter.	$C \in \{10^i\}_{i=0}^{3}$
	σ: RBF kernel coefficient.	$\sigma \in \{10^i\}_{i=-2}^{2}$
kNN	k: Number of nearest neighbors to compute distance.	$k \in \{3, 5, 9, 13, 15, 25, 40\}$.

Source: Florencio V *et al.* (2020).

High-dimension datasets III

Table 13: Black-box experiment results for RMSE and R^2 in high dimension datasets.

		FULL-MLM	RF	*k*NN	SVR	LW-MLM-1	LW-MLM-2	LW-MLM-3
RMSE	RS	**318.89**	736.70	396.97	575.75	**378.09**	**378.35**	**381.64**
	RC	**47.34**	84.26	89.38	**39.46**	52.44	51.82	53.20
	CC	**871.78**	1558.03	1859.21	1243.90	985.36	935.71	989.50
	RB	**0.59**	0.83	0.80	0.74	**0.59**	**0.59**	**0.59**
R^2	RS	**0.93**	0.61	0.53	0.77	**0.90**	**0.90**	**0.90**
	RC	**0.92**	0.73	0.70	**0.94**	**0.90**	**0.90**	**0.90**
	CC	**0.90**	0.66	0.51	0.78	**0.88**	**0.89**	**0.88**
	RB	**0.67**	0.15	0.18	0.32	**0.67**	**0.67**	**0.67**

Source: Florencio V *et al.* (2020).

Computational complexity

Table 14: MLM variant comparison in terms of memory requirement, training cost, and out-of-sample prediction cost. We recall that N stands for the cardinality of the training set, while M is the number of RPs ($N \geq M$).

MODEL	MEMORY	TRAINING	PREDICTION
Full-MLM	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Random-MLM	$\mathcal{O}(NM)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M)$
Rank-MLM	$\mathcal{O}(NM)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M)$
LW-MLM	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(M)$

Source: Florencio V *et al.* (2020).

[Proposal#4] CCLW-MLM: When LW-MLM meets CCIS

The Class-Corner "Lightweight" Minimal Learning Machine

Class-Corner Lightweight Minimal Learning Machine (CCLW-MLM)

- Integrate CCIS during the LW-MLM learning phase as regularization by class-corner nearness;

Regularization by class-corner nearness

First, let us define $\mathcal{PS}$ be the set yielded by CCIS, we then define the Nearest Corner Distance $\mathrm{NCD}(\cdot)$ of a given sample $\mathbf{x}$ as:

$$\mathrm{NCD}_{\mathcal{PS}}(\mathbf{x}) = \min \left\{ ||\mathbf{x} - \mathbf{x}_j||_2 \right\}, \forall \mathbf{x}_j \in \mathcal{PS}. \tag{27}$$

Now, we define the ceiling cost as the maximum distance of a query sample to the class-corners for all samples in $\mathcal{D}$:

$$\zeta = \max \left\{ \mathrm{NCD}_{\mathcal{PS}}(\mathbf{x}_i) \right\}, \forall \mathbf{x}_i \in \mathcal{D}, \tag{28}$$

so that we can derive the cost by class-corner nearness of a given sample as:

$$\varsigma(\mathbf{x}) = \zeta - \mathrm{NCD}_{\mathcal{PS}}(\mathbf{x}), \tag{29}$$

where finally we can employ $P_{i,i} = \varsigma(\mathbf{x}_i)$ in CCLW-MLM.

We carried out two types of experiments to evaluate different aspects of our proposal. Our goal is to investigate how CCLW-MLM behaves with respect to the following aspects: accuracy and sparseness. Each of the following experiments addresses such concerns. Also, we highlight all experiments were performed using a Mac Mini with an 3.6 GHz Intel Core i3 Quad-Core, 8 GB of RAM, and running macOS Catalina 10.15.6 with Python 3.7.3.

Experiment Setup

- **The typical black-box assessment on some datasets**: Here, we are interested in assessing CCLW-MLM against some of the state-of-the-art MLM variants that employ regularization through accuracy and sparseness in a typical black-box test fashion on the same "toy-size" datasets.
- **Empirical decision boundary quality assessment**: This experiment investigates visually the quality of solutions produced by the MLM variants that employ regularization by empirically evaluating the decision boundaries.

Typical black-box assessment for CCLW-MLM

We compared CCLW-MLM against the Full-MLM, Rank-MLM, and Random-MLM. In this comparison, we report the average accuracy (ACC) and sparseness via $||\mathbf{B}||_{\mathcal{F}}$ over 30 independent realizations. The hyperparameter setting for each variation, including ours, is presented in Table 15.

Table 15: MLM variants and their hyperparameters configurations.

MODEL	DESCRIPTION	HYPERPARAMETER SPACE
Full-MLM	None.	None.
Random-MLM	K: Number of RPs chosen randomly from a range through grid search and random cross-validation.	$k \sim \mathcal{U}(0.05, 0.5)$ so that $K = \lfloor k \times N \rfloor$.
Rank-MLM	λ : Regularization parameter optimized by grid search and cross-validation.	$\log(\lambda) \sim \mathcal{U}(\log(1e-3), \log(1e2))$.
CCLW-MLM	$\mathbf{P}$: Diagonal matrix with values from the complements of their closeness to the corners in $\mathcal{PS}$.	See Equation (29).

Typical black-box assessment for CCLW-MLM

We report the average results for 30 independent realizations for these 10 toy-size datasets in Table 16.

Table 16: Description of the datasets: name, acronym, input dimensionality and number of training and test samples.

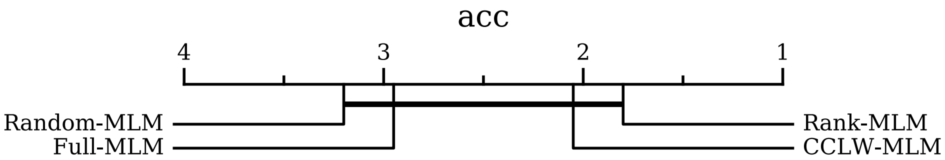
DATASET	FULL-MLM	Random-MLM		Rank-MLM		CCLW-MLM	
	ACC	ACC	$\|\|\mathbf{B}\|\|_{\mathcal{F}}$	ACC	$\|\|\mathbf{B}\|\|_{\mathcal{F}}$	ACC	$\|\|\mathbf{B}\|\|_{\mathcal{F}}$
BAN	0.88 ± 0.00	0.89 ± 0.00	0.86 ± 0.00	0.90 ± 0.00	0.99 ± 0.00	0.90 ± 0.00	0.99 ± 0.00
BCW	0.97 ± 0.00	0.97 ± 0.01	0.77 ± 0.00	0.97 ± 0.00	0.05 ± 0.00	0.96 ± 0.00	0.92 ± 0.00
GER	0.74 ± 0.00	0.74 ± 0.01	0.82 ± 0.00	0.74 ± 0.00	0.43 ± 0.00	0.74 ± 0.00	0.87 ± 0.00
HAB	0.74 ± 0.00	0.74 ± 0.01	0.93 ± 0.00	0.76 ± 0.00	0.99 ± 0.00	0.75 ± 0.00	0.97 ± 0.00
HEA	0.82 ± 0.00	0.82 ± 0.01	0.74 ± 0.00	0.81 ± 0.00	0.83 ± 0.00	0.82 ± 0.00	0.72 ± 0.00
ION	0.91 ± 0.00	0.88 ± 0.03	0.56 ± 0.00	0.91 ± 0.00	0.30 ± 0.00	0.90 ± 0.00	0.76 ± 0.00
PID	0.72 ± 0.00	0.74 ± 0.01	0.73 ± 0.00	0.74 ± 0.00	0.78 ± 0.00	0.76 ± 0.00	0.94 ± 0.00
RIP	0.88 ± 0.00	0.89 ± 0.00	0.91 ± 0.00	0.89 ± 0.00	0.99 ± 0.00	0.89 ± 0.00	0.99 ± 0.00
TMN	1.00 ± 0.00	0.96 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.40 ± 0.00	1.00 ± 0.00	0.47 ± 0.00
VCP	0.79 ± 0.00	0.82 ± 0.02	0.75 ± 0.00	0.83 ± 0.00	0.68 ± 0.00	0.83 ± 0.00	0.91 ± 0.00

*The values regarding the standard deviation are not zero, but very small ones.

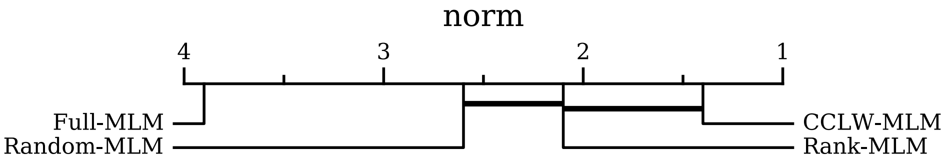
Again, we adopted the scaled factor using **B from Full-MLM because **D** and **Δ** are full-rank matrices, and it does not employ any regularization factor. Thus, $||\mathbf{B}||_{\mathcal{F}}$ from Full-MLM serves as a superior limit value for it. This can be recognized by the following property of matrix norms: $||\mathbf{B}||_{\mathcal{F}} = ||\mathbf{D}^{-1}\mathbf{\Delta}||_{\mathcal{F}} \geq ||\mathbf{D}^{-1}||_{\mathcal{F}}||\mathbf{\Delta}||_{\mathcal{F}}$.

Typical black-box assessment for CCLW-MLM

Figure 26: Critical difference plots with respect to the accuracy (a) and sparsity rankings (b) from Table 16.



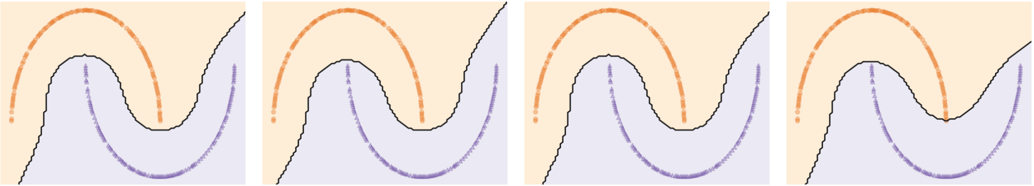
(a) Accuracy rankings.



(b) $||\mathbf{B}||_{\mathcal{F}}$ rankings.

Typical black-box assessment for CCLW-MLM

Figure 28: Results for Two Moon dataset.



(a) Two Moon for Full-MLM.

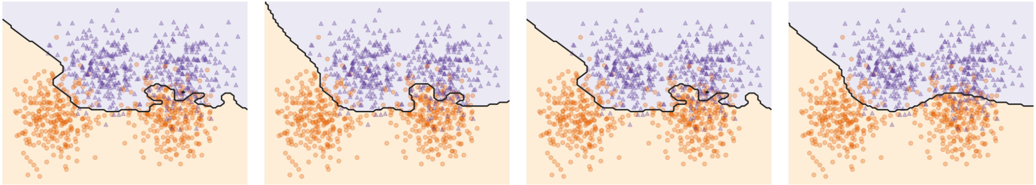
(b) Two Moon for Random-MLM.

(c) Two Moon for Rank-MLM.

(d) Two Moon for CCLW-MLM.

Typical black-box assessment for CCLW-MLM

Figure 30: Results for Two Moon dataset.



(a) Ripley for Full-MLM.

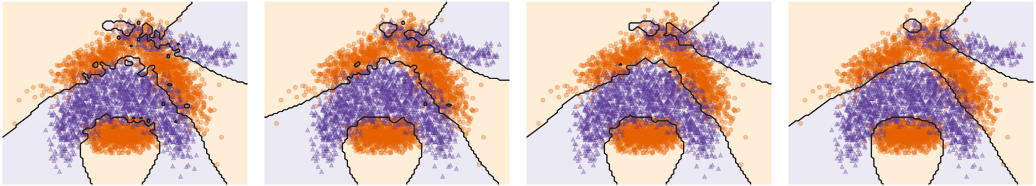
(b) Ripley for Random-MLM.

(c) Ripley for Rank-MLM.

(d) Ripley for CCLW-MLM.

Typical black-box assessment for CCLW-MLM

Figure 32: Results for Two Moon dataset.



(a) Banana for Full-MLM.

(b) Banana for Random-MLM.

(c) Banana for Rank-MLM.

(d) Banana for CCLW-MLM.

Concluding Remarks

Concluding Remarks I

Conclusion

- The contribution presented in this thesis is four-part and investigated the model complexity reduction in two Instance-based learners, namely: The Least-Squares Support vector machine and the Minimal Learning Machine.
- The common idea behind all the solutions is to reduce the complexity in Instance-based learners from instance selection, treating it as a regularization task. Thus, excluding our first contribution, an instance selection algorithm itself, we modified the design of such LSSVM and MLM algorithms to embed such a complexity reduction.
- Chapter 3 presented the first well-succeded attempt to reduce the LSSVM complexity by selecting class-corner data points. Based on FAST, an image processing algorithm for corner detection, this thesis's first contribution is formulated and named Class Corner Instance Selection (CCIS). It deals with the instance selection problem by choosing data points near the boundary of classes. From that, we extend a pruned and reduced set LSSVM model, after named CC-LSSVM.

Concluding Remarks II

Conclusion

- In Chapter 4, we dealt with reducing complexity in MLMs after applying regularization by sample. Such a formulation derived our third contribution, named Lightweight Minimal Learning Machine, which took advantage of such a strategy to learn in a restricted hypothesis space and generate a faster model for predicting regression tasks.
- Chapter 5 revisited the LW-MLM and formulated this thesis's final contribution by combining CCIS to LW-MLM. We named it Class-Corner Lightweight Minimal Leaning Machine because it deals with classification tasks straightforwardly.
- We carried out some experiments to evaluate each contribution's different aspects, investigating how they behave concerning the following aspects: the prediction error, the goodness-of-fit of estimated vs. measured values, the model complexity, the influence of the parameters, and the learned models' empirical visual analysis.
- Even though our contributions strongly rely on distance computations, thus being suffering from the Dimensionality curse, they consistently outperformed the other models in artificial and real-world scenarios. This thesis's apparent unfolding is to directly apply metric learning methods to derive more algorithms with consistent hypotheses.

References

References I

ALENCAR, A. S. C.; CALDAS, W. L.; GOMES, J. P. P.; JUNIOR, A. H. de S.; AGUILAR, P. A. C.; RODRIGUES, C.; FRANCO, W.; CASTRO, M. F. de; ANDRADE, R. M. C. MLM-Rank: A Ranking algorithm based on the Minimal Learning Machine. In: Soc Brasileira Comp SBC; Univ Federal do Rio Grande do Norte. **2015 BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS 2015)**. Rio Grande do Norte, 2015. p. 305–309. ISBN 978-1-5090-0016-6. 4th Brazilian Conference on Intelligent Systems (BRACIS), Natal, BRAZIL, NOV 04-07, 2015.

CARVALHO, B. P. R.; BRAGA, A. P. IP-LSSVM: A two-step sparse classifier. **PATTERN RECOGNITION LETTERS**, 30, n. 16, p. 1507–1515, DEC 1 2009. ISSN 0167-8655.

CORTES, C.; VAPNIK, V. Support-Vector Networks. **MACHINE LEARNING**, 20, n. 3, p. 273–297, SEP 1995. ISSN 0885-6125.

COVER, T.; HART, P. Nearest Neighbor Pattern Classification. **IEEE TRANSACTIONS ON INFORMATION THEORY**, 13, n. 1, p. 21+, 1967. ISSN 0018-9448.

DIAS, M. L. D.; FREIRE, A. L.; JUNIOR, A. H. S.; NETO, A. R. da R.; GOMES, J. P. P. Sparse minimal learning machines via l(1/2) norm regularization. In: Brazilian Comp Soc. **2018 7TH BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS)**. São Paulo, 2018. p. 206–211. ISBN 978-1-5386-8023-0. 7th Brazilian Conference on Intelligent Systems (BRACIS), IBM Res, Sao Paulo, BRAZIL, OCT 22-25, 2018.

FLORENCIO V, J. A.; OLIVEIRA, S. A. F.; GOMES, J. P. P.; NETO, A. R. R. A new perspective for Minimal Learning Machines: A lightweight approach. **NEUROCOMPUTING**, 401, p. 308–319, AUG 11 2020. ISSN 0925-2312.

References II

GARCIA, S.; DERRAC, J.; CANO, J. R.; HERRERA, F. Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**, 34, n. 3, p. 417–435, MAR 2012.

GOMES, J. P. P.; SOUZA JR., A. H.; CORONA, F.; NETO, A. R. R. A Cost Sensitive Minimal Learning Machine for Pattern Classification. In: Arik, S and Huang, T and Lai, WK and Liu, Q (Ed.). **NEURAL INFORMATION PROCESSING, PT I**. Istanbul, Turkey: [s.n.], 2015. (Lecture Notes in Computer Science, 9489), p. 557–564. ISBN 978-3-319-26532-2; 978-3-319-26531-5. ISSN 0302-9743. 22nd International Conference on Neural Information Processing (ICONIP), Istanbul, TURKEY, NOV 09-12, 2015.

JIAO, L.; BO, L.; WANG, L. Fast sparse approximation for least squares support vector machine. **IEEE TRANSACTIONS ON NEURAL NETWORKS**, 18, n. 3, p. 685–697, MAY 2007. ISSN 1045-9227.

JÚNIOR, A. H. de S. **Regional Models and Minimal Learning Machines for Nonlinear Dynamic System Identification**. Tese (Doutorado) — Federal University of Ceará, Fortaleza, Brazil, October 2014.

JUNIOR, A. H. de S.; CORONA, F.; BARRETO, G. A.; MICHE, Y.; LENDASSE, A. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. **NEUROCOMPUTING**, 164, p. 34–44, SEP 21 2015. ISSN 0925-2312. 12th International Work-Conference on Artificial Neural Networks (IWANN), Puerto de la Cruz, SPAIN, JUN 12-14, 2013.

KAHAKI, S. M. M.; NORDIN, M. J.; ASHTARI, A. H. Contour-Based Corner Detection and Classification by Using Mean Projection Transform. **SENSORS**, 14, n. 3, p. 4126–4143, MAR 2014. ISSN 1424-8220.

References III

MACKAY, D. Bayesian Interpolation. **Neural Computation**, 4, n. 3, p. 415–447, MAY 1992. ISSN 0899-7667.

MALL, R.; SUYKENS, J. A. K. Very Sparse LSSVM Reductions for Large-Scale Data. **IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS**, 26, n. 5, p. 1086–1097, MAY 2015. ISSN 2162-237X.

NIEWIADOMSKA-SZYNKIEWICZ, E.; MARKS, M. Optimization schemes for wireless sensor network localization. **IntINTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND COMPUTER SCIENCE**, Walter de Gruyter & Co., Hawthorne, NJ, USA, v. 19, n. 2, p. 291–302, jun. 2009. ISSN 1641-876X. Disponível em: <https://doi.org/10.2478/v10006-009-0025-3>.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial: Tradução da 3a Edição**. Elsevier Brasil, 2017. ISBN 9788535251418. Disponível em: <https://books.google.com.br/books?id=BsNeAwAAQBAJ>.

OLIVEIRA, S. A. F.; GOMES, J. P. P.; NETO, A. R. R. Sparse Least-Squares Support Vector Machines via Accelerated Segmented Test: A dual approach. **NEUROCOMPUTING**, 321, p. 308–320, DEC 10 2018. ISSN 0925-2312.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: Leonardis, A and Bischof, H and Pinz, A (Ed.). **COMPUTER VISION - ECCV 2006 , PT 1, PROCEEDINGS**. Graz, Austria, 2006. (LECTURE NOTES IN COMPUTER SCIENCE, 1), p. 430–443. ISBN 3-540-33832-2. ISSN 0302-9743.

References IV

THEODORIDIS, S. **Machine Learning: A Bayesian and Optimization Perspective**. Elsevier Science, 2020. ISBN 9780128188040. Disponível em: <https://books.google.com.br/books?id=l-nEDwAAQBAJ>.

TIPPING, M. Sparse Bayesian learning and the relevance vector machine. **JOURNAL OF MACHINE LEARNING RESEARCH**, 1, n. 3, p. 211–244, SUM 2001. ISSN 1532-4435.

YANG, L.; YANG, S.; ZHANG, R.; JIN, H. Sparse least square support vector machine via coupled compressive pruning. **NEUROCOMPUTING**, 131, p. 77–86, MAY 5 2014. ISSN 0925-2312.

Thank you for your attention!