Journal of Machine Learning Research 21 (2020) 1-29 Submitted 9/19; Revised 8/20; Published 10/20

Minimal Learning Machine: Theoretical Results and Clustering-Based Reference Point Selection

Joonas Hämäläinen JOONAS.K.HAMALAINEN@JYU.FI
University of Jyvaskyla, Faculty of Information Technology
P.O. Box 35, FI-40014 University of Jyvaskyla, Finland

Alisson S. C. Alencar ALENCAR.ALISSON@LIA.UFC.BR
Federal University of Ceará—UFC, Department of Computer Science
Fortaleza-CE, Brazil

Tommi Kärkkäinen TOMMI.KARKKAINEN@JYU.FI
University of Jyvaskyla, Faculty of Information Technology
P.O. Box 35, FI-40014 University of Jyvaskyla, Finland

César L. C. Mattos CESARLINCOLN@DC.UFC.BR
Federal University of Ceará—UFC, Department of Computer Science
Fortaleza-CE, Brazil

Amauri H. Souza Júnior AMAURIHOLANDA@IFCE.EDU.BR
Federal Institute of Education, Science and Technology of Ceará—IFCE
Department of Computer Science, Maracanaú-CE, Brazil

João P. P. Gomes JPAULO@DC.UFC.BR
Federal University of Ceará—UFC, Department of Computer Science
Fortaleza-CE, Brazil

Editor: Amos Storkey

Abstract

The Minimal Learning Machine (MLM) is a nonlinear, supervised approach based on learning linear mapping between distance matrices computed in input and output data spaces, where distances are calculated using a subset of points called reference points. Its simple formulation has attracted several recent works on extensions and applications. In this paper, we aim to address some open questions related to the MLM. First, we detail the theoretical aspects that assure the MLM's interpolation and universal approximation capabilities, which had previously only been empirically verified. Second, we identify the major importance of the task of selecting reference points for the MLM's generalization capability. Several clustering-based methods for reference point selection in regression scenarios are then proposed and analyzed. Based on an extensive empirical evaluation, we conclude that the evaluated methods are both scalable and useful. Specifically, for a small number of reference points, the clustering-based methods outperform the standard random selection of the original MLM formulation.

Keywords: Minimal Learning Machine, Interpolation, Universal Approximation, Clustering, Reference Point Selection

1. Introduction

Machine learning techniques can be roughly categorized as unsupervised and supervised, depending on whether the learning data comprises only input data or a complete set of input-output pairs (Shalev-Shwartz and Ben-David, 2014). In terms of target data, semi-supervised learning typically lies somewhere between these extremes (Gan et al., 2013), and active (Aggarwal et al., 2014) or incremental (Losing et al., 2018) learning techniques acquire the desired outputs during model construction incrementally, on a need-to-know basis. A key concept in unsupervised learning, especially clustering, is the distance or dissimilarity between two observations or an observation-metaobservation (e.g., cluster prototype) pair (Reddy and Vinzamuri, 2013). Currently, supervised learning extensively uses deep structures with multiple layers of weights and stochastic optimization in training (Hubara et al., 2017).

The distance-based supervised methods provide a methodological middle ground and link between unsupervised and supervised learning. Recent examples of such methods include the Minimal Learning Machine (de Souza Junior et al., 2015) and the Extreme Minimal Learning Machine (Kärkkäinen, 2019). These methods' core learning construct is distance regression, based on the dissimilarity between observations. Hence, nonlinear regression and classification can be performed for all entities whose dissimilarity can be metrically defined. During learning, incremental use of the so-called reference points, together with the solution of the corresponding distance-based linear system, is necessary, without any optimization procedure (Kärkkäinen, 2019). Note that such distance-based supervised techniques also enable direct utilization of metric learning techniques as part of their construction (e.g., Kulis, 2013).

The MLM's increasing popularity can be explained by its simple formulation, easy implementation, and promising results in several applications (Mesquita et al., 2017a; Coelho et al., 2014; Marinho et al., 2017, 2018; Pihlajamäki et al., 2020). Apart from the MLM's applications, many studies from 2015 to 2020 have sought to improve and augment the MLM's basic form in order to handle missing values (Mesquita et al., 2015, 2017b) and outliers (Gomes et al., 2017), perform ensemble learning (Mesquita et al., 2017a) and semi-supervised learning (Caldas et al., 2018), speed up its computations (Florêncio et al., 2020; Mesquita et al., 2017a; Marinho et al., 2016), and include a reject option in classification tasks (de Oliveira et al., 2016).

1.1 Prior Work on Distance-Based Learning

Radial Basis Function Networks (RBFN) (Powell, 1987; Broomhead and Lowe, 1988) popularized the use of distance in training data as part of neural network models. Usually, the distance in RBFN is further transformed with a nonlinear activation function, but early papers analyzing the technique also explicated the use of a linear, distance-based kernel (Poggio and Girosi, 1990; Park and Sandberg, 1991).

The actual development of dissimilarity-based machine learning techniques was advanced by Pekalska and Duin (2001), who proposed using a "global classifier defined on the similarities to a small set of prototypes, called the representation set." This representation set is the set of reference points in MLM parlance. Moreover, similarly to Step 1 in MLM (see Section 2), dissimilarities and the corresponding distance matrix between objects in the representation and training sets were used by Pekalska and Duin (2001), who applied the regularized linear normal density classifier. A linear classifier model based on dissimilarities was then proposed by Pekalska et al. (2001), who estimated parameters similarly to the SVM by solving a linear programming problem for the separating hyperplane in binary classification. Fisher's Linear Discriminant was used for distance-based spectral classification by Paclık and Duin (2003).

According to Balcan et al. (2008), use of the Euclidean distance function corresponds to the trivial identity kernel and to the corresponding scaled similarity function (Balcan et al., 2008, Definition 1). Let us refer to this as the Euclidean kernel below. This means that the distance transformation in the MLM introduces the famous kernel trick, where the size of the implicit space coincides with the number of reference points. However, because the whole construction of the kernelized learning in MLM occurs in the distance space, this formulation differs from the SVM or kernel-perceptron and from the approaches with dissimilarity kernels previously proposed by Pekalska and Duin (2001), Pekalska et al. (2001), Paclık and Duin (2003), Pekalska et al. (2006), Pekalska and Duin (2008), Chen et al. (2009), and Wang et al. (2009).

Closely related work to ours—again, in the context of Step 1 of the MLM—is by Zerzucha et al. (2012), who used the complete Euclidean dissimilarity matrix with the partial least squares method. Fuzzy clustering and leave-one-out cross-validation were suggested for the identification of the most informative subset of data (i.e., reference point selection) and for the reduced Euclidean distance matrix.

Feature selection combined with a distance-based classification of imbalanced data was considered by Zhang et al. (2015), who used Naive Bayes, instance-based nearest neighbor, Random Forest, Multilayer Perceptron, and Logistic Regression from WEKA as classifiers. Note that SVM is the dominant (only practically) method used with distance-based kernels for time series classification (Abanda et al., 2019). A dissimilarity-based method with Random Forest as a classifier was proposed by Cao et al. (2019). A recent review on various dissimilarity-based approaches was provided by Costa et al. (2020).

In conclusion, the use of distances, dissimilarities, or proximities in supervised learning is not new (e.g., Balcan et al. 2008; Chen et al. 2009; Schleif and Tino 2015). As summarized by Chen (2010), the most straightforward utilization of distance calculations is using the pairwise distances as features of a predictive model. Indeed, this utilization is part of the MLM, which is additionally characterized by reference point selection, genuine distance regression, and the solution of a multilateration problem. Hence, the whole learning framework with MLM differs from earlier work in the field, as depicted, for example, by Pekalska and Duin (2001), Paclık and Duin (2003), Pekalska et al. (2006), Wang et al. (2007), Pekalska and Duin (2008), Balcan et al. (2008), Nguéma and Saint-Pierre (2008), Chen et al. (2009), Chen (2010), and Schleif and Tino (2015).

1.2 The Importance of Reference Point Selection

In the MLM, reference points are a subset of training points, and they are used to build the distance matrices that are a key component of the MLM's induction process. In the original MLM formulation, the reference points were randomly selected. As empirically demonstrated by de Souza Junior et al. (2015), a poor choice of reference points can damage the MLM's generalization capability. This phenomenon is even more likely to occur when the number of reference points is small (de Souza Junior et al., 2015). An example of the effects of different reference point selection strategies is provided in Section 4.2.

Also, a large number of proposals are available to improve the behaviour of the distance-based methods using reference point selection (reference points are also referred to as prototypes or landmarks). Pekalska et al. (2006) considered prototype/reference point selection with Bayesian classifiers. They concluded that a set of few, evenly distributed centers provided better classification results (faster with higher accuracy) than the use of all training examples.

Later suggestions toward this direction were provided by Plasencia-Calaña et al. (2014, 2017), again in the form of finding a small set of prototypes. A well-spread set of diverse reference points was also suggested as part of the similarity-based learning framework by Kar and Jain (2011).

Dias et al. (2018) proposed a strategy to select reference points based on identifying of the class boundaries in a binary classification problem. The proposal prohibited selecting any point as a reference point from a subset of points in the class boundary area. A similar objective was pursued by Florêncio et al. (2018), who identified such a region using fuzzy c-means. Maia et al. (2018) used a sparse regression method to build a linear mapping between distance matrices. They selected the reference points according to the resulting non-zero coefficients obtained by the linear model.

Even if the previous works on reference point selection have led to more compact models with better generalization, the existing efforts have only focused on classification problems. Additionally, none of these works have presented any theoretical results that can explain the impact of choosing reference points in a general setting.

1.3 Contributions

In the present work, we advance the research field described above by: i) presenting proof of the MLM's interpolation capability when all training points are used as reference points; ii) demonstrating the MLM's universal approximation capability—even in scenarios where reference point selection is considered; and iii) proposing and analyzing several reference point selection strategies for regression problems based on elements of clustering methods.

When we choose clustering-based approaches, our basic hypothesis suggests, a set of well-spread reference points in the data space will improve the MLM's performance compared to random selection. We validated this paper's empirical contributions through computational experiments with 15 regression data sets.

The remainder of this paper is organized as follows. Section 2 presents the MLM's basic formulation. Section 3 details our theoretical contributions to the MLM's interpolation and generalization capabilities. Section 4 describes clustering-based methodologies for reference point selection. Section 5 presents a comprehensive set of experiments to evaluate clustering-based methodologies for reference point selection. Finally, Section 6 concludes the paper.

2. Minimal Learning Machine

As previously discussed, the MLM is a distance-based supervised machine learning method. The basic algorithm (de Souza Junior et al., 2013, 2015) comprises two main steps: i) regression estimation using the distance-based kernel and ii) distance-based interpolation of a new output. For clarity, we describe these two steps below.

Let $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ be a set of training inputs, where $\boldsymbol{x}_i \in \mathbb{R}^P$ and $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$ is the set of the corresponding outputs, for $\boldsymbol{y}_i \in \mathbb{R}^L$, respectively. Moreover, we define the set of (input) reference points $\mathcal{R} = \{\boldsymbol{r}_k\}_{k=1}^K$ as a non-empty subset of $\mathcal{X}$, $\mathcal{R} \subseteq \mathcal{X}$, and let $\mathcal{T} = \{\boldsymbol{t}_k\}_{k=1}^K$ refer to the outputs of the corresponding reference inputs, that is, $\boldsymbol{r}_k \mapsto \boldsymbol{t}_k$.

Next, we define two distance matrices, $\boldsymbol{D}_x \in \mathbb{R}^{N\times K}$ and $\boldsymbol{D}_y \in \mathbb{R}^{N\times K}$, using the Euclidean distance $\|\cdot\|$ as follows:

$$\boldsymbol{D}_x = \left[\|\boldsymbol{x}_i - \boldsymbol{r}_k\|\right] \quad i = 1, \dots, N,\ k = 1, \dots, K, \tag{1}$$

$$\boldsymbol{D}_y = \left[\|\boldsymbol{y}_i - \boldsymbol{t}_k\|\right] \quad i = 1, \dots, N,\ k = 1, \dots, K. \tag{2}$$

The key idea for the MLM's first step is the assumption of a regression model between the distance matrices: $\boldsymbol{D}_y = g(\boldsymbol{D}_x) + \boldsymbol{E}$, where $\boldsymbol{E}$ denotes the residuals/error in this transformation. Assuming that the unknown regression model is linear in form, its transformation matrix $\boldsymbol{B} \in \mathbb{R}^{K\times K}$ can be estimated using the well-known ordinary least squares formulation, as follows:

$$\boldsymbol{B} = \left(\boldsymbol{D}_x^T \boldsymbol{D}_x\right)^{-1} \boldsymbol{D}_x^T \boldsymbol{D}_y. \tag{3}$$

The linear mapping represented by the matrix $\boldsymbol{B}$, obtained in Eq. (3), is the MLM's first step.

For the second step, let $\tilde{\boldsymbol{x}}$ be a new input vector whose output must be estimated. Hence, based on the distance regression model from the first step, we seek the corresponding output $\tilde{\boldsymbol{y}}$, satisfying

$$\|\tilde{\boldsymbol{y}} - \boldsymbol{t}_k\| \approx \delta_k \quad \forall k = 1, \dots, K, \tag{4}$$

where

$$\boldsymbol{\delta} = \left[\|\tilde{\boldsymbol{x}} - \boldsymbol{r}_k\|\right]_{k=1}^K \boldsymbol{B}.$$

The solution to the multilateration problem in Eq. (4) can also be obtained using the least-squares formulation by letting

$$\tilde{\boldsymbol{y}}^* = \operatorname{argmin} \mathcal{J}(\tilde{\boldsymbol{y}}), \quad \text{where} \quad \mathcal{J}(\tilde{\boldsymbol{y}}) = \sum_{k=1}^K \left(\|\tilde{\boldsymbol{y}} - \boldsymbol{t}_k\|^2 - \boldsymbol{\delta}_k^2\right)^2. \tag{5}$$

As stated by de Souza Junior et al. (2015), many possible solvers exist for Eq. (5). In the original formulation, the MLM solves the output estimation step using a nonlinear optimization algorithm. Such an algorithm is used to find the point that minimizes the double-quadratic error between the estimated distance and the real distance, calculated on each candidate point. However, we want to verify whether, when the distances are perfectly estimated, the point's position can be recovered without error. To that end, we follow an alternative formulation of the multilateration problem, called the Localization Linear System (LLS), detailed in Appendix A. This formulation provides an efficient output estimation method. The LLS method computes the output position by solving a linear system. An output prediction algorithm for the MLM with LLS is depicted in Algorithm 1. Substitution "$\leftarrow$ [ ]" referes to the removal of an element from a vector.

Algorithm 1 MLM output prediction with LLS

```
Input: input $\tilde{\boldsymbol{x}}$, distance regression model $\boldsymbol{B}$, reference points $\mathcal{R}$ and $\mathcal{T}$.
Output: predicted output $\tilde{\boldsymbol{y}}$.
 1: $\boldsymbol{d}_{\tilde{\boldsymbol{x}}} \leftarrow \left[\|\tilde{\boldsymbol{x}} - \boldsymbol{r}_k\|\right]_{k=1}^{K}$
 2: $\boldsymbol{\delta} \leftarrow \boldsymbol{d}_{\tilde{\boldsymbol{x}}}\boldsymbol{B}$                          // Predict distances in the output space
 3: $i^* \leftarrow rand(\{1, \dots, K\})$
 4: $\boldsymbol{t}^*, \delta^* \leftarrow \boldsymbol{t}_{i^*}, \boldsymbol{\delta}(i^*)$
 5: $\mathcal{T}, \boldsymbol{\delta}(i^*), K \leftarrow \mathcal{T}\backslash\{\boldsymbol{t}_{i^*}\}, [\,], K-1$   // Remove BAN from the set of reference points
 6: $\boldsymbol{b}, \boldsymbol{A}$
 7: for $i \in \{1, \dots, K\}$ do
 8:     $\boldsymbol{b}(i) \leftarrow \frac{1}{2}(\delta^{*2} + \|\boldsymbol{t}^* - \boldsymbol{t}_i\|^2 - \boldsymbol{d}_{\tilde{\boldsymbol{x}}}(i)^2)$
 9:     $\boldsymbol{A}(i,:) \leftarrow (\boldsymbol{t}_i - \boldsymbol{t}^*)^T$
10: $\boldsymbol{\theta} \leftarrow solve(\boldsymbol{A}\boldsymbol{\theta} = \boldsymbol{b})$                    // Solve a linear system of equations
11: $\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{\theta} + \boldsymbol{t}^*$
```

In summary, the LLS solves a system in the form $\boldsymbol{A}\boldsymbol{\theta} = \boldsymbol{b}$. The coefficient matrix $\boldsymbol{A}$ is constructed based on all but one reference point, named the "benchmark-anchor-node" (BAN), and each row i is given by the difference between the i-th reference point and the BAN. The vector $\boldsymbol{\theta}$ is a simple translation of the target position. The vector $\boldsymbol{b}$ is computed from the estimated distances between the target point and the reference points, as well as the distance from the BAN itself to the other reference points.

In Algorithm 1, a linear system of equations in Step 10 is usually overdetermined. An approximate solution can be obtained from the ordinary least squares (OLS) method with a computational cost of $\mathcal{O}(L^2K)$. Step 1 has a computational cost of $\mathcal{O}(KP)$. Usually, Step 2 is, computationally, the most expensive step, and it determines the asymptotic behavior of the computational complexity, $\mathcal{O}(K^2)$, when $K >> P$ and $K >> L$. Therefore, models with a reduced number of reference points can lead to a significant computational time reduction for the MLM prediction with the LLS when the input and output space dimensions are small compared to K.

Nyström approximation Initially, one could draw similarities between the MLM formulation and the methods that consider a Nyström approximation for Gram matrices (Williams and Seeger, 2001; Drineas and Mahoney, 2005; Sun et al., 2015). Such methods are based on the approximation $\boldsymbol{K} \approx \boldsymbol{C}\boldsymbol{W}^{\dagger}\boldsymbol{C}^T$, where $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, $\boldsymbol{W} \in \mathbb{R}^{K\times K}$, $\boldsymbol{C} \in \mathbb{R}^{N\times K}$, and $K \ll N$.

However, the exact least squares solution $\boldsymbol{B} = (\boldsymbol{D}_x^T\boldsymbol{D}_x)^{-1}\boldsymbol{D}_x^T\boldsymbol{D}_y$ in the MLM's first step differs. Indeed, for $K < N$, the distance matrices $\boldsymbol{D}_x \in \mathbb{R}^{N\times K}$ and $\boldsymbol{D}_y \in \mathbb{R}^{N\times K}$ are rectangular, and we cannot obtain the same solution by directly applying the standard Nyström approach. We confirm the latter statement by considering the full distance matrices $\boldsymbol{\Delta}_x \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\Delta}_y \in \mathbb{R}^{N\times N}$, which correspond to the solution $\boldsymbol{B} = \boldsymbol{\Delta}_x^{-1}\boldsymbol{\Delta}_y$. By considering a Nyström representation for $\boldsymbol{\Delta}_x$, we would obtain $\boldsymbol{B} \approx (\boldsymbol{C}\boldsymbol{W}^{\dagger}\boldsymbol{C}^T)^{-1}\boldsymbol{\Delta}_y$. The inverse $(\boldsymbol{C}\boldsymbol{W}^{\dagger}\boldsymbol{C}^T)^{-1}$ may exist only for $K = N$, the only case where we recover the least squares solution by choosing, for instance, $\boldsymbol{C} = \boldsymbol{W} = \boldsymbol{\Delta}_x$.

Nevertheless, the vectors used to build the matrix $\boldsymbol{W}$ in a Nyström approximation, usually called ”landmarks,” can be seen as analogous to the reference points in the MLM. In the literature on the Nyström method, clustering algorithms are well known as a sensible strategy for choosing landmarks (Zhang et al., 2008; Zhang and Kwok, 2010; Kumar et al., 2012; Oglic and Gärtner, 2017; Pourkamali-Anaraki et al., 2018). This observation encourages us to also pursue a clustering approach to select the MLM's reference points.

3. MLM Theoretical Results

In this section, we detail some of the MLM's theoretical guarantees. These results are divided into two subsections: ”Interpolation Theory” and ”Universal Approximation Capability.”

3.1 Interpolation Theory

We show that the MLM can interpolate data in two steps. First, we show that the distance matrix D_x, constructed using all points in the available data as reference points, is invertible. According to Eq. (3), and given that $D_x^T = D_x$ when all data points act as reference points, the distances can be estimated accurately. In the second step, we prove that—under certain conditions that will be described—the estimation of the output will recover the original points' position with zero error.

3.1.1 Inverse of distance matrices

In the MLM's training phase, we need to solve a linear system whose coefficient matrix is given by the distances between the points of the data set and the reference points—that is, a matrix D_x, such that $d_{i,j}$ is given by $d(\boldsymbol{x_i}, \boldsymbol{r_j})$, the distance between the i-th point of the training set and the j-th reference point. If we consider the specific case in which all points in the data set are reference points, then the coefficient matrix is a square matrix of order equal to the number of training points N. We rearrange the points so that $x_i = r_i, \forall i \in \{1, \cdots, N\}$, and the matrix of coefficients is such that each element $d_{i,j}$ is given by $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A matrix with this characteristic is formally called a ”distance matrix.” To find an exact solution, we must show that every distance matrix admits an inverse.

The invertibility of the distance matrix was first demonstrated by Micchelli (1986); Auer (1995) offered a simplified proof. The main result is given by the following theorem:

Theorem 1 *Given a distance matrix $\boldsymbol{D}$ computed from a set of N distinct points, the determinant of $\boldsymbol{D}$ is positive if N is odd and negative if N is even; specifically, $\boldsymbol{D}$ is invertible.*

With this result, we can guarantee that, when the distance matrix in the input space is multiplied by the coefficient matrix obtained in the MLM training, the result is the distance matrix in the output space—without any error.

3.1.2 Condition for the perfect estimation of the multilateration

The result of the previous subsection is important since it shows that the MLM can recover the distances in the output space between the reference points and the training data with zero errors. However, this result is not sufficient evidence to suggest that the MLM is capable of interpolating any data set. For that claim, we must prove the model's ability to estimate the output—that is, to retrieve the points' position in the output space from the perfectly estimated distances.

Solving the LLS accurately is only possible when the coefficient matrix is non-singular. This condition is not necessarily true of any set of points. Indeed, Theorem 2 below shows that the matrix is invertible when the reference points, including the BAN, form an independent affine set.

Theorem 2 (perfect estimation with the multilateration) *Given a linearly independent spanning set $\boldsymbol{v}_1 \dots \boldsymbol{v}_R \in \mathbb{R}^S$ and $\boldsymbol{v} \in \mathbb{R}^S$. If $\boldsymbol{v}$ is not an affine combination of $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_R\}$, then the set of vectors $\boldsymbol{v}_1 - \boldsymbol{v}, \boldsymbol{v}_2 - \boldsymbol{v}, \dots, \boldsymbol{v}_R - \boldsymbol{v}$ is linearly independent.*

Proof:

Suppose that $\boldsymbol{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_R\}$ is a linearly independent spanning set, $\boldsymbol{v}$ is not an affine combination of $\boldsymbol{V}$, and $\boldsymbol{V}' = \{\boldsymbol{v}_1 - \boldsymbol{v}, \dots, \boldsymbol{v}_R - \boldsymbol{v}\}$ is linearly dependent. There then exists $\mu_1, \dots, \mu_R$, not all equal to zero, such that:

$$\begin{aligned}
\sum_{i=1}^{R} \mu_i(\boldsymbol{v}_i - \boldsymbol{v}) &= 0 \\
\sum_{i=1}^{R} \mu_i(\boldsymbol{v}_i - \sum_{j=1}^{R} \lambda_j \boldsymbol{v}_j) &= 0 \\
\sum_{i=1}^{R} \mu_i \boldsymbol{v}_i - \sum_{i=1}^{R} \mu_i \sum_{j=1}^{R} \lambda_j \boldsymbol{v}_j &= 0 \\
\sum_{i=1}^{R} \mu_i \boldsymbol{v}_i - \sum_{i=1}^{R} \lambda_i \boldsymbol{v}_i \sum_{j=1}^{R} \mu_j &= 0 \\
\sum_{i=1}^{R} (\mu_i \boldsymbol{v}_i - \lambda_i \boldsymbol{v}_i \sum_{j=1}^{R} \mu_j) &= 0 \\
\sum_{i=1}^{R} \underbrace{(\mu_i - \lambda_i \sum_{j=1}^{R} \mu_j)}_{\theta_i} \boldsymbol{v}_i &= 0 \\
\sum_{i=1}^{R} \theta_i \boldsymbol{v}_i &= 0. \qquad (6)
\end{aligned}$$

Since $\boldsymbol{V}$ is LI, Eq. (6) can only be satisfied when all θ_i are equal to zero, which means $\mu_i = \lambda_i \sum \mu_j, \forall i$. If $\sum \mu_j = 0$, we have $\mu_i = 0, \forall i$; however, this cannot be true since we assume that μ_i are not all zero. Assuming, then, that $\sum \mu_j \neq 0$, we have $\lambda_i = \frac{\mu_i}{\sum \mu_j}$; however, this gives $\sum \lambda_i = 1$. Since we assume that $\boldsymbol{v}$ is not an affine combination of $\boldsymbol{V}$, we arrive at a contradiction and conclude the proof.

The above theorem shows that the multilateration results in the point's exact position in the output space when we choose S linearly independent points from the training set and another point (the BAN) that is not an affine combination of the other points. Since the number of training points is usually much larger than the dimension of the output, this is usually possible.

3.2 Universal Approximation Property

We will now verify an important theoretical result of the MLM: its universal approximation capability. This result is divided in two parts: one part is for the distance estimation error after the linear transformation, and the other part is for the multilateration estimation error when recovering the output position. This result will clarify that the MLM can be used to approximate arbitrary functions.

3.2.1 Upper bound for distance estimation error

To show that the distance estimation error computed by the MLM is bounded, we will use a result presented by Park and Sandberg (1993), who showed that a Radial Basis Function (RBF) network is a universal approximation. The result is summarized by Theorem 3, as follows:

Theorem 3 (RBF universal approximation) *Let $\kappa : \mathbb{R}^r \to \mathbb{R}$ be a nonzero integrable function, such that κ is continuous and radially symmetric with respect to the Euclidean norm. Then, the family S_κ is dense in the space of continuous $\mathbb{R}$-valued maps defined on any compact subset of $\mathbb{R}^r$ with respect to the norm $||.||_\infty$, where S_κ is the family of RBF networks with kernel function κ, given by*

$$q(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \kappa \left(\frac{\boldsymbol{x} - \boldsymbol{z_i}}{\sigma_i} \right).$$

This theorem shows that an RBF network can approximate a significant set of functions with an arbitrarily small error. For the MLM, we can resort to this result by considering that the desired output of the data set is the distance to the reference points in the output space. With that modification, we must show that the MLM can be described in the RBF network formalism, which ensures that the MLM can estimate the distances to the reference points in the output with an arbitrarily small error.

We will first take the centroids of the RBF as the MLM's reference points. The function κ then takes the Euclidean norm, given by $\kappa(\frac{\boldsymbol{x}-\boldsymbol{z_i}}{\sigma_i}) = ||\frac{\boldsymbol{x}-\boldsymbol{z_i}}{\sigma_i}||$. The presented RBF formulation has a parameter σ_i that does not appear in the MLM. However, if a combination w_i, σ_i satisfies the property, we can calculate $\bar{w}_i = \frac{w_i}{\sigma_i}$ and get the same result. Finally, both the RBF and the MLM apply a linear regression to compute the output, so we can state that the weights $\boldsymbol{w}$ of the RBF are equivalent to the coefficients of matrix $\boldsymbol{B}$ in Eq. (3). Thus, we conclude the proof that the error of the MLM-estimated distances in the output space can be arbitrarily small.

3.2.2 Upper bound for the multilateration prediction error

In the previous section, we showed that the MLM can provide a good estimate of the distances in the output space. However, the MLM requires an additional step to compute the output: the multilateration. This section shows that the multilateration estimation error is bounded.

As Hu et al. (2016) showed, an upper bound is found for the error of multilateration, given by the method detailed in Appendix A. This work was conducted in the context of mobile autonomous robot localization, and it differed from the MLM in some ways. In summary, Hu et al. (2016) aimed to locate a mobile robot based on estimated distances for some fixed points of known locations, called "anchor points." Both the distance estimates and the anchor point locations themselves may present noise. Thus, in that context, the upper bound for the multilateration error is expressed by Theorem 4, as follows:

Theorem 4 (upper bound for the LLS error) *An LLS constructed is described in Eq.* (9) *in Appendix A and is expressed*

$$\hat{\boldsymbol{A}}\boldsymbol{\theta} = \hat{\boldsymbol{b}},$$

where $\hat{\boldsymbol{A}} = \boldsymbol{A} + \Delta\hat{\boldsymbol{A}}$ *is a matrix constructed by the anchors' positions,* $\boldsymbol{A}$ *represents the anchor nodes' precise position,* $\Delta\hat{\boldsymbol{A}}$ *is the anchors' coordinate errors,* $\hat{\boldsymbol{b}} = \boldsymbol{b} + \Delta\hat{\boldsymbol{b}}$ *is a vector collection of the anchors' positions and the measurement data,* $\boldsymbol{b}$ *denotes the noiseless measurement data, and* $\Delta\hat{\boldsymbol{b}}$ *represents the noise of the measurement data. The ratio between the estimated coordinate* $\hat{\boldsymbol{y}}$ *and the true coordinate* $\boldsymbol{y}$ *satisfies*

$$\frac{||\hat{\boldsymbol{y}}||}{||\boldsymbol{y}||} \leq \psi(1+\alpha)(1+\beta),$$

where

$$\psi = ||\hat{\boldsymbol{A}}^{\dagger}||||\hat{\boldsymbol{A}}||,$$

$$\alpha = \frac{||\Delta\hat{\boldsymbol{A}}||}{||\hat{\boldsymbol{A}}||},$$

$$\beta = \frac{1}{|||\hat{\boldsymbol{b}}||_2/||\Delta\hat{\boldsymbol{b}}||_2 - 1|}.$$

An MLM analogy can be made with the presented context by considering that the robot's location is the desired output and the anchor points' locations are the locations of the reference points in the output space. Distance estimates from the robot to the anchor points are given by the MLM's output before the multilateration step.

Theorem 4 presents an upper bound for the multilateration error. However, the MLM's characteristics allow us to tighten the bound. First, we consider that the reference points' location is accurate, which means $\Delta\boldsymbol{A} = \boldsymbol{0}$; thus, $\alpha = 0$. In addition, we saw in the previous section that the MLM distance estimate errors can be arbitrarily small. This means $\Delta\boldsymbol{b} \to \boldsymbol{0}$, which implies $\beta \to 0$. Thus, we can present the following corollary:

Corollary 5 *The error of the MLM multilateration step is bounded by*

$$\frac{||\hat{\boldsymbol{y}}||}{||\boldsymbol{y}||} \leq \psi(1+\beta) = \mathcal{U},$$

where

$$\psi = ||\hat{\boldsymbol{A}}^{\dagger}||||\hat{\boldsymbol{A}}||,$$
$$\beta = \frac{1}{|||\hat{\boldsymbol{b}}||_2/||\Delta\hat{\boldsymbol{b}}||_2 - 1|}.$$

In addition, since we have $\beta \to 0$, we have $\mathcal{U} \to \psi$.

The result of this corollary shows that the ratio between the returned and the desired output is bounded. We will develop this relation to show that the distance between the returned and the desired output is also bounded:

$$\begin{aligned}
d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2 =& (\hat{\boldsymbol{y}} - \boldsymbol{y})^T(\hat{\boldsymbol{y}} - \boldsymbol{y}) \\
d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2 =& ||\hat{\boldsymbol{y}}||^2 + ||\boldsymbol{y}||^2 - 2\boldsymbol{y}^T\hat{\boldsymbol{y}} \\
d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2 =& ||\hat{\boldsymbol{y}}||^2 + ||\boldsymbol{y}||^2 - 2||\boldsymbol{y}||||\hat{\boldsymbol{y}}|| \cos\alpha \\
\frac{d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2}{||\boldsymbol{y}||^2} =& \frac{||\hat{\boldsymbol{y}}||^2}{||\boldsymbol{y}||^2} + \frac{||\boldsymbol{y}||^2}{||\boldsymbol{y}||^2} - 2\cos\alpha\frac{||\boldsymbol{y}||||\hat{\boldsymbol{y}}||}{||\boldsymbol{y}||^2} \\
\frac{d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2}{||\boldsymbol{y}||^2} =& \left(\frac{||\hat{\boldsymbol{y}}||}{||\boldsymbol{y}||}\right)^2 + 1 - 2\cos\alpha\frac{||\hat{\boldsymbol{y}}||}{||\boldsymbol{y}||} \\
\frac{d(\hat{\boldsymbol{y}}, \boldsymbol{y})^2}{||\boldsymbol{y}||^2} \leq& (\mathcal{U})^2 + 1 - 2\cos\alpha(\mathcal{U}).
\end{aligned} \tag{7}$$

We can, therefore, conclude that if the norm $||\boldsymbol{y}||$ of the target output is bounded, the distance $d(\hat{\boldsymbol{y}}, \boldsymbol{y})$ between the desired output and the output estimated by the multilateration is also bounded.

3.3 Discussion

Corollary 5 indicates that the upper bound of the multilateration error depends on matrix $\boldsymbol{A}$, which is itself associated with the reference points used to compute the distances. This observation indicates that we can tighten the bound for certain choices of reference points, thereby reducing the output estimation error limit. This idea was previously demonstrated empirically (Dias et al., 2018; Florêncio et al., 2018; Maia et al., 2018). In the present work, we have now theoretically motivated a non-random selection for the reference points. We assess this motivation by performing the comprehensive computational experiments detailed in the next sections, focusing on clustering-based approaches.

4. Clustering-Based Reference Point Selection

In this section, we evaluate four clustering-based methods in the reference point selection problem. These methods include two nondeterministic and two deterministic ones. A general algorithm for the selection of clustering-based reference points is depicted in Algorithm 2. All the methods are based on a common strategy where the selection of reference points is performed only in the input space. The corresponding points (indices) are simply selected as output references. Therefore, below, we consider only the input space when describing the proposed methods.

```
Algorithm 2 Clustering-based selection of reference points
Input: input points $\mathcal{X}$, output points $\mathcal{Y}$, and number of reference points $K$
Output: reference points $\mathcal{R}$ and $\mathcal{T}$
1: Cluster $\mathcal{X}$ to $K$ clusters
2: Select cluster prototype from each cluster
3: Select $\mathcal{R}$ according to the cluster prototypes from $\mathcal{X}$
4: Select $\mathcal{T}$ corresponding to indices of $\mathcal{R}$ from $\mathcal{Y}$
```

Method	**Based on**	**Deterministic**	**Type**	**Complexity**
RS-K-means++	K-means++ initialization	No	Partitional	$\mathcal{O}(N)$
RS-K-medoids++	K-means++ initialization and K-medoids clustering	No	Partitional	$\mathcal{O}(N)$
RS-UPGMA	Aggloremerative clustering	Yes	Hierarchical	$\mathcal{O}(N^2)$
RS-maximin	Maximin clustering initialization	Yes	Partitional	$\mathcal{O}(N)$

Table 1: Summary of the evaluated reference point selection approaches.

4.1 Methods

The K-means++ initialization method (Arthur and Vassilvitskii, 2007) is among the most popular methods of K-means initialization. The first method we evaluate is the use of the K-means++ initialization with the Euclidean distance for reference point selection. See Hämäläinen et al. (2017) for a description of the algorithm. We will refer to this approach as reference point selection with K-means++ (RS-K-means++).

The second evaluated approach begins by running the K-means++ initialization with the Euclidean distance, and then it refines the initial prototypes with Lloyd's algorithm (Lloyd, 1982) until convergence. Finally, the closest observation to each final prototype (medoid) is selected as the reference point. These closest points then establish the set of selected reference points. This method is referred to as RS-K-medoids++. Both RS-K-medoids++ and RS-K-means++ are nondeterministic methods because of the random sampling of the initial prototypes, which are based on the Euclidean distance-constructed probability distribution (see Hämäläinen et al. 2017 and the articles therein).

The unweighted pair group method with the arithmetic mean (UPGMA; Sokal, 1958) is an agglomerative clustering algorithm that starts clustering from the initial state, where each point forms one cluster. Then, in each step, the two clusters with the smallest average distance between the cluster members are joined together. The third evaluated method utilizes UPGMA on the data, and then it computes the mean prototypes for each cluster; finally, it again selects the closest point to the prototype as a reference point. Similar to RS-K-medoids++, those closest points construct the set of selected reference points. We refer to this method as RS-UPGMA.

The fourth evaluated method is based on a maximin clustering initialization algorithm (Gonzalez, 1985). The original method starts with a random initial point and then picks each new point, similar to the K-means++ method. However, unlike K-means++, the point with the farthest distance from the closest already selected point is chosen as a new point. Our modification of the maximin first selects the closest point to the data mean as the first point, conceiving of the whole algorithm as completely deterministic. This approach is referred to as RS-maximin. We emphasize that the latter two approaches, RS-UPGMA and RS-maximin, are deterministic.

One justification for selecting this specific set of clustering methods is the highly different amounts of separation between the selected reference points (see Figure 2 in Appendix B). Random selection involves the smallest amount of separation among the reference points, and the RS-maximin method involves the largest amount; RS-K-means++, RS-K-medoids++, and RS-UPGMA interpolate between these two extremes. Plenty of clustering methods are available; the methods evaluated here are straightforward and easy to implement. Moreover, the MLM has only one hyperparameter, the number of reference points K to be selected, which the methods keep unchanged.

A summary of the evaluated approaches is shown in Table 1, where the time complexities are also presented with respect to the number of training observations N. RS-K-means++, RS-K-medoids++, and RS-maximin have linear time complexity. The UPGMA has quadratic complexity (Gronau and Moran, 2007); therefore, the complexity of RS-UPGMA is also quadratic, since the post-processing after the UPGMA clustering step has linear time complexity. Since the MLM training phase has a time complexity of $\mathcal{O}(K^2N)$ (de Souza Junior et al., 2015), a reference point selection method with a linear computational cost (with respect to N) and an ability to build an accurate model with a small K is highly desirable.

4.2 Motivation for Reference Point Selection

To illustrate reference point selection's effects in terms of the MLM's accuracy, we generated a nonlinear synthetic data set (6,240 observations, 1 input variable, 1 output variable) with varying density. Input values are drawn from four highly different density intervals. Corresponding output values are given by a cubic function with Gaussian noise. The MLM was trained with the Random and RS-maximin methods when $K = 10$. In addition, we trained the full MLM variant. Figure 1 illustrates that the Random method selects reference points from high-density regions, which causes the MLM to have a very low accuracy in low-density regions. RS-maximin also selects reference points from the low-density regions, which clearly improves accuracy. Selecting reference points from near the data cloud boundaries improves the MLM regression model's extrapolation capability, as also illustrated by Hämäläinen (2018). A straightforward approach to also cover low-density regions is to include all the data points as a reference points; however, this approach can lead to overfitting and very large MLM models. On the other hand, overfitting seems to rarely be a problem for multidimensional input spaces, based on this paper's results and the works of Florêncio et al. (2020), Hämäläinen and Kärkkäinen (2020), Kärkkäinen (2019), and Pihlajamäki et al. (2020). Especially in classification problems, small noise on the class boundaries, as characterized in Figure 1 (right), might not affect classification accuracy.

5. Experiments and Results

In this section, we show empirical evidence with an extensive set of data sets how the MLM's generalization accuracy can be improved in regression problems with clustering-based reference point selection. We used the Random selection as a baseline for the clustering-based methods.

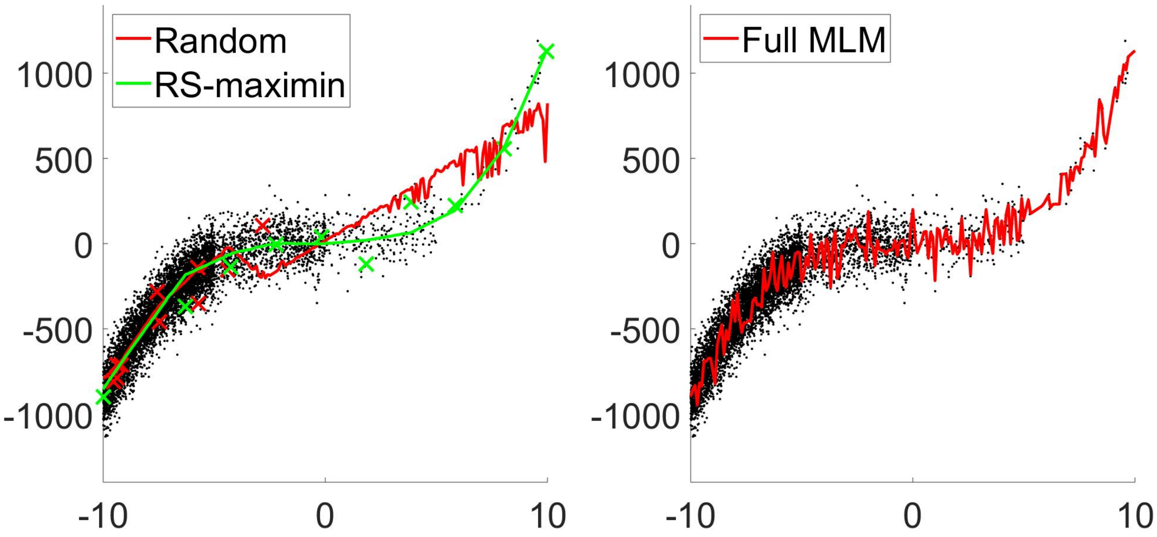

Figure 1: Illustration of the effects for different reference point selection strategies. Selected reference points are marked with crosses with the corresponding color of the fitted curve.

Data set	# Observations	# Features
Auto Price (AP)	159	15
Servo (SRV)	167	4
Breast Cancer (BC)	194	32
Computer Hardware (CHA)	209	6
Boston Housing (BH)	506	13
Forest Fires (FF)	517	12
Stocks (STC)	950	9
S1 (S1)	1,000	2
Bank (BNK)	4,499	8
Ailerons (ALR)	7,129	5
Computer Activity (CA)	8,192	12
Elevators (ELV)	9,517	6
Combined Cycle Power Plant (CCP)	9,568	4
California Housing (CH)	20,640	8
Census (CNS)	22,784	8

Table 2: Characteristics of the data sets used in the experiments.

5.1 Experimental Setup

We selected 13 real data sets and two synthetic data sets (S1, BNK) to evaluate the reference point selection methods. The selected data sets are summarized in Table 2. All data sets had one-dimensional output values. The S1 data set was modified for a regression task. We randomly selected 1,000 observations from the original S1 data, scaled their values to the range $[0, 1]$ and then computed the output values $f(x_1, x_2)$ with the function $\sin(2\pi x_1) + \sin(2\pi x_2)$. The original S1 data set is available at `http://cs.uef.fi/sipu/datasets/`. The remaining data sets are available at `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html` and at `http://archive.ics.uci.edu/ml/index.php`.

For a more rigorous comparison, we performed model selection and assessment as follows. We divided the original data sets into train-validation-test sets and performed cross-validation (see, e.g., Friedman et al., 2001, Chapter 7). More precisely, we used the 3-DOB-SCV (Moreno-Torres et al., 2012) approach to divide each data set into a training set and a test set. Therefore, the test set was forced to approximate the same distribution as the training set, making the comparison more reliable if concept drift is not considered. Because we focused only on regression tasks, we used DOB-SCV as a one-class case (Hämäläinen, 2018; Hämäläinen and Kärkkäinen, 2016). Moreover, we archived three training sets and three test sets for each data set, respectively, with sizes of 2/3 and 1/3 of the number of observations. In training, we used the 10-DOB-SCV approach to select the optimal number of reference points. Hence, 18/30 of the number of observations were used to train the model and 2/30 of the number of observations were used to compute the validation error. Therefore, we have a two-level division of the data sets.

We evaluated the models' quality using the root mean square error (RMSE). In addition to the validation error, a test error was also computed for all 10-DOB-SCV training sets, resulting in 10 test RMSEs for each training set and 30 test RMSEs for the overall data set. For more interpretable results, we expressed the number of selected reference points relatively:

$$K_{rel} = 100\frac{K}{N}, \tag{8}$$

where N is the number of observations in the training data. In training, the number of reference points K_{rel} varied in the range of $[5, 100]$, with a step size of 5. We used the LLS method for output prediction (Algorithm 1). To solve the linear system of equations in the MLM implementation, we utilized MATLAB's *mldivide*-function. We scaled all training observations to the range $[0, 1]$. All the experiments were conducted in a MATLAB environment.

5.2 Results for Optimal K

Table 3 shows the median test RMSE and the optimal number of reference points. The optimal number of reference points was selected based on the smallest mean validation RMSE. The symbol $**$ indicates a statistically significant difference between test RMSEs, based on a Kruskal-Wallis H test with a significance level of 0.05. The symbols $*$, †, ‡, §, and ‖ denote that a method has a statistically significantly smaller RMSE in pairwise comparison to Random, RS-K-means++, RS-K-medoids++, RS-UPGMA, and RS-maximin, respectively. In the pairwise comparisons, the significance level was also set to 0.05. The Kruskal-Wallis H test assumes equal variances for groups; therefore, we tested the equality of the variances with a Brown-Forsythe test. Based on this test, the variances related to optimal K results were equal for all data sets. The best median test RMSE and the set of the smallest number of reference points (with respect to the mean value) are in boldface for each data set. Note that Table 3 includes three optimal K values for each method, since

	Random		**RS-K-means++**		**RS-K-medoids++**		**RS-UPGMA**		**RS-maximin**	
Data set	**RMSE**	**K$_{\mathbf{rel}}$**	**RMSE**	**K$_{\mathbf{rel}}$**	**RMSE**	**K$_{\mathbf{rel}}$**	**RMSE**	**K$_{\mathbf{rel}}$**	**RMSE**	**K$_{\mathbf{rel}}$**
AP	0.0640(85)	75, 90, 80	0.0600(77)	65, 80, 95	0.0597(79)	**45, 100, 55**	**0.0583**(**68**)	90, 95, 100	**0.0583**(**68**)	90, 100, 95
SRV	0.0920(77)	**90, 100, 90**	0.0910(76)	90, 100, 95	0.0908(76)	95, 100, 95	**0.0899**(75)	100, 100, 95	0.0913(**74**)	95, 100, 100
BC	0.2678(78)	10, 20, 30	0.2674(85)	5, 25, 15	0.2664(76)	**10, 15, 10**	0.2680(77)	10, 15, 15	**0.2647**(**63**)	15, 20, 10
CHA**	0.0483(93)	80, 15, 80	0.0461(98)	**50, 15, 25**	0.0431(66)†	25, 65, 55	**0.0403**(**52**)*†	20, 95, 20	0.0421(68)	65, 55, 15
BH	0.0728(81)	75, 100, 100	**0.0717**(**72**)	95, 100, 100	0.0725(**75**)	95, 85, 100	**0.0717**(**75**)	**85, 85, 75**	0.0718(**75**)	85, 95, 100
FF	**0.0557**(74)	40, 10, 10	0.0564(82)	5, 45, 10	0.0559(75)	30, 10, 55	0.0566(**71**)	**5, 5, 15**	0.0566(76)	**5, 15, 5**
STC	**0.0227**(**73**)	100, 100, 100	0.0228(77)	**95, 100, 95**	0.0228(74)	100, 95, 100	**0.0227**(**75**)	**100, 100, 90**	0.0228(78)	**95, 100, 95**
S1	**0.0051**(**76**)	100, 100, 100	0.0053(76)	75, 100, 95	0.0052(78)	80, 90, 70	**0.0051**(75)	**80, 90, 65**	0.0052(**74**)	85, 100, 95
BNK**	0.0514(95)	90, 95, 90	0.0515(93)	85, 100, 70	0.0509(81)	100, 55, 60	0.0490(59)*†	95, 5, 10	**0.0481**(**51**)*†	**5, 10, 10**
ALR	**0.0417**(**66**)	10, 10, 10	0.0418(71)	5, 10, 15	0.0418(69)	10, 5, 15	0.0420(87)	5, 10, 20	0.0420(85)	**5, 10, 5**
CA	**0.0288**(75)	90, 100, 100	**0.0288**(75)	90, 75, 75	**0.0288**(**72**)	60, 85, 75	0.0289(79)	**70, 65, 60**	**0.0288**(77)	80, 95, 70
ELV	0.0554(76)	**5, 5, 5**	**0.0553**(**74**)	**5, 5, 5**	0.0554(75)	**5, 5, 5**	**0.0553**(**75**)	**5, 5, 5**	**0.0553**(76)	5, 5, 10
CCP	0.0478(**73**)	85, 100, 100	**0.0476**(**73**)	90, 70, 95	0.0480(77)	70, 85, 95	0.0482(81)	**55, 80, 95**	0.0480(74)	75, 80, 95
CH	0.1137(77)	**70, 85, 70**	**0.1134**(75)	80, 80, 95	0.1135(76)	80, 80, 95	0.1135(75)	100, 95, 100	0.1136(**74**)	100, 100, 90
CNS	0.0605(83)	20, 35, 30	0.0605(80)	15, 20, 20	0.0603(75)	15, 25, 25	**0.0599**(**64**)	15, 25, 15	0.0602(76)	**10, 30, 10**
Rank / K_{rel}^{avg}	5(54) / 64.44		4(48) / 59.56		3(44) / 58.44		2(40) / **55.22**		**1(39)** / 56.33	

Table 3: RMSE for the optimal K.

we used the 3-DOV-SCV approach in the experiments. Rounded Kruskal-Wallis scores are shown inside the brackets, and the best scores are in boldface. Data set—wise ranking of the methods was calculated from the raw Kruskal-Wallis scores. Based on these rankings, the final ranking of the methods is shown at the bottom of Table 3. In addition, the average K_{rel} is also shown at the bottom of Table 3 for each method.

Based on Table 3, RS-UPGMA and RS-maximin performed equally well in the final ranking, while RS-K-medoids++ and RS-K-means++ performed similarly. In terms of the final ranking and the model size (K_{rel}), Random had the worst performance and the deterministic methods RS-UPGMA and RS-maximin performed best. In general, clustering-based methods give sparser models that reduce computational costs and space requirements. In addition, the clustering-based models have better generalization ability. Based on the Kruskal-Wallis test, the methods differ statistically significantly for the CHA and BNK data sets in favor of the deterministic methods. For the BNK data set, RS-maximin builds the MLM model with only $K_{rel} = \{5, 10, 10\}$, while Random must select almost the entire data set as reference points ($K_{rel} = \{90, 95, 90\}$) and still has a clearly larger RMSE error. Reducing K_{rel} from 90 to 10 reduces space requirements for the distance regression model coefficient matrix by 98.77%. For large data sets where $N >> P$ and $N >> L$, this coefficient matrix size determines the full MLM model's ($K_{rel} = 100$) space complexity.

The best K selection, based on the smallest mean validation RMSE, is dubious for some of the data sets since the model's complexity of the model is not taken into account. For example, for a large data set, if increasing K_{rel} from 50 to 100 leads to only marginal improvement in the mean validation RMSE, then the model with higher K and smaller mean validation RMSE is selected. For example, for the S1 data set, RS-maximin already achieves the fulll MLM error level when $K_{rel} = 20 - 40$ (see Tables 6 and 7). For future work, room for improvement remains in this respect.

Data set	**Random**	**RS-K-means++**	**RS-K-medoids++**	**RS-UPGMA**	**RS-maximin**
AP**	0.1083(89)	0.1082(83)	0.1052(86)	0.0954(66)	**0.0829**(**54**)*‡
SRV**	**0.1921**(**57**)$^{\|}$	0.2024(70)	0.2088(86)	0.2011(71)	0.2132(94)
BC	0.2672(**68**)	0.2684(80)	**0.2670**(71)	0.2707(86)	0.2671(72)
CHA	0.0697(82)	**0.0593**(**57**)	0.0659(80)	0.0682(78)	0.0608(81)
BH	0.1171(70)	0.1194(80)	0.1141(84)	0.1141(80)	**0.1099**(**63**)
FF	0.0572(81)	**0.0565**(76)	0.0568(81)	0.0566(**67**)	0.0566(73)
STC**	0.0521(121)	0.0478(87)*	0.0457(45)$^{*\dagger\|}$	**0.0449**(**41**)$^{*\dagger\|}$	0.0477(84)*
S1**	0.0366(128)	0.0285(91)*	0.0270(78)*	0.0241(56)*†	**0.0199**(**25**)*†‡§
BNK**	0.0645(103)	0.0584(87)	0.0670(117)	0.0499(42)*†‡	**0.0491**(**30**)*†‡
ALR	**0.0417**(**70**)	0.0418(68)	0.0419(77)	0.0420(83)	0.0420(79)
CA**	0.0341(120)	0.0320(70)*	**0.0314**(54)*§	0.0324(85)*	**0.0314**(**48**)*§
ELV	0.0554(77)	**0.0553**(75)	0.0554(76)	**0.0553**(76)	**0.0553**(**73**)
CCP	0.0528(88)	0.0526(79)	0.0526(73)	0.0525(70)	**0.0522**(**67**)
CH**	**0.1201**(**44**)$^{\ddagger\S\|}$	0.1208(58)$^{\S\|}$	0.1219(75)	0.1230(101)	0.1232(99)
CNS	0.0622(91)	0.0613(77)	0.0614(79)	**0.0607**(68)	**0.0607**(**62**)
Rank	5(55)	2(42)	4(52)	3(43)	**1(33)**

Table 4: RMSE for $K_{rel} = 5$.

Data set	**Random**	**RS-K-means++**	**RS-K-medoids++**	**RS-UPGMA**	**RS-maximin**
AP	0.0930(83)	0.0916(85)	0.0856(78)	0.0838(70)	**0.0762**(**62**)
SRV**	0.1479(64)	0.1500(60)	0.1651(91)	0.1693(108)	**0.1468**(**54**)
BC	0.2674(77)	0.2667(72)	0.2668(77)	0.2692(85)	**0.2655**(**67**)
CHA**	0.0613(105)	0.0542(92)	0.0496(81)	**0.0428**(**43**)*†‡	0.0448(57)*†
BH	**0.0994**(**70**)	0.1003(76)	0.1017(75)	0.1018(81)	0.1011(75)
FF	0.0568(78)	0.0568(79)	0.0572(80)	0.0567(70)	**0.0565**(**72**)
STC**	0.0395(122)	0.0375(102)	0.0359(59)*†	**0.0353**(**45**)*†	0.0359(50)*†
S1**	0.0188(123)	0.0140(92)*	0.0135(81)*	0.0109(60)*†	**0.0078**(**23**)*†‡§
BNK**	0.0589(103)	0.0570(93)	0.0588(97)	0.0487(47)*†	**0.0481**(**37**)*†
ALR	**0.0417**(**65**)	0.0418(72)	0.0418(63)	0.0420(86)	0.0422(92)
CA**	0.0316(121)	0.0302(73)*	**0.0299**(**61**)*	0.0305(71)*	0.0301(52)*
ELV	0.0557(85)	0.0555(83)	0.0555(76)	0.0555(76)	**0.0554**(**66**)
CCP	0.0516(88)	0.0516(78)	0.0515(72)	0.0515(71)	**0.0511**(**68**)
CH**	**0.1177**(**44**)$^{\S\|}$	0.1185(59)$^{\S\|}$	0.1193(74)	0.1212(100)	0.1212(101)
CNS	0.0612(89)	0.0605(78)	0.0605(80)	**0.0601**(67)	0.0602(**64**)
Rank	5(58)	4(52)	3(46)	2(41)	**1(28)**

Table 5: RMSE for $K_{rel} = 10$.

5.3 Results for Fixed K

Tables 4–7 show the test RMSEs. They are similar to Table 3, but with a fixed number of reference points. Variances for the error distributions are not equal for SRV ($K_{rel} = 5, 10, 20$), CHA ($K_{rel} = 10, 20, 40$), BH ($K_{rel} = 10, 20$), FF ($K_{rel} = 10$), STC ($K_{rel} = 5, 10, 20, 40$), S1 ($K_{rel} = 5, 10, 20$), CA ($K_{rel} = 5, 20, 40$), or ELV ($K_{rel} = 10$), based on the Brown-Forsythe test of group variances. Therefore, the Kruskal-Wallis results are questionable in these cases. However, the methods' ordering can still be compared.

As expected, based on the final ranking, all the proposed methods have better RMSE than Random when the number of reference points is small to moderate ($K_{rel} = 5, 10, 20, 40$, Tables 4–7). RS-K-means++ have better RMSEs than RS-K-medoids++ for $K_{rel} = 5$. Thus, refinement of the reference points with K-means does not seem beneficial for the small K_{rel}. In contrast to $K_{rel} = 10, 20$, accuracy improves with K-means refinement. In general, the RS-maximin method obtained the best RMSE in the comparison. The RS-UPGMA results are fairly similar to RS-maximin results for $K_{rel} = 20, 40$. Therefore, running the whole clustering (not only the initialization step) seems to work better for higher K values. For $K_{rel} = 20$, RS-UPGMA is the best approach, based on the final ranking.

Data set	**Random**	**RS-K-means++**	**RS-K-medoids++**	**RS-UPGMA**	**RS-maximin**
AP	0.0858(87)	0.083(82)	0.0794(72)	0.0775(68)	**0.0738**(**68**)
SRV	0.1226(73)	0.1244(**71**)	**0.1213**(**71**)	0.1239(82)	0.1225(80)
BC	0.2671(80)	**0.2651**(**70**)	0.2659(80)	0.2655(73)	0.2655(74)
CHA**	0.0595(111)	0.0475(95)	0.0443(72)*	**0.0403**(**45**)*†	0.0430(55)*†
BH	0.0913(82)	0.0875(79)	0.0890(81)	**0.0836**(**58**)	0.0857(78)
FF	0.0565(80)	0.0564(76)	0.0568(82)	**0.0560**(**70**)	0.0565(**70**)
STC**	0.0324(129)	0.0304(95)*	0.0296(83)*	0.0283(42)*†‡	**0.0277**(**29**)*†‡
S1**	0.0113(128)	0.0082(88)*	0.0083(83)*	0.0069(49)*†‡	**0.0057**(**31**)*†‡
BNK**	0.0560(105)	0.0539(89)	0.0531(85)	0.0490(53)*†‡	**0.0487**(**46**)*†‡
ALR**	**0.0419**(**62**)	**0.0419**(**69**)	0.0420(68)	0.0422(88)	0.0423(91)
CA**	0.0305(117)	0.0294(70)*	**0.0293**(**66**)*	**0.0293**(**61**)*	0.0294(64)*
ELV	0.0561(84)	0.0561(82)	0.0560(79)	0.0558(69)	**0.0557**(**63**)
CCP	0.0504(86)	0.0501(78)	0.0501(**66**)	**0.0498**(68)	0.0504(79)
CH**	**0.1159**(**45**)$^{\S\Vert}$	0.1165(63)$^{\S\Vert}$	0.1167(69)$^{\Vert}$	0.1192(99)	0.1194(102)
CNS	0.0609(87)	0.0605(80)	0.0603(77)	**0.0599**(**64**)	0.0602(69)
Rank	5(63)	4(49)	3(45)	**1(32)**	2(36)

Table 6: RMSE for $K_{rel} = 20$.

Data set	**Random**	**RS-K-means++**	**RS-K-medoids++**	**RS-UPGMA**	**RS-maximin**
AP	0.0749(89)	0.0704(79)	0.0701(79)	**0.0647**(**57**)*	0.0682(73)
SRV	0.1072(79)	0.1072(81)	**0.1014**(**73**)	0.1045(72)	0.1058(72)
BC	0.2679(80)	0.2689(80)	0.2682(77)	0.2666(71)	**0.2662**(**62**)
CHA	0.0478(94)	**0.0411**(**62**)*	0.0430(74)	0.0428(70)	0.0436(77)
BH**	0.0843(97)	0.0818(86)	0.0800(73)	**0.0769**(**61**)*	**0.0769**(**61**)*
FF	0.0559(80)	0.0566(75)	0.0564(82)	0.0601(79)	**0.0545**(**61**)
STC**	0.0276(131)	0.0260(97)*	0.0256(75)*	**0.0247**(**35**)*†‡	0.0248(39)*†‡
S1**	0.0073(121)	0.0060(78)	0.0059(72)	0.0054(54)	**0.0052**(**53**)
BNK**	0.0536(105)	0.0527(89)	0.0512(76)	0.0500(58)*	**0.0495**(**50**)*†
ALR	**0.0423**(**63**)	0.0425(75)	0.0426(79)	0.0426(85)	0.0424(77)
CA**	0.0291(102)	0.0289(74)	0.0289(69)*	**0.0288**(**66**)*	0.0289(67)*
ELV	0.0572(78)	0.0570(79)	0.0570(81)	**0.0568**(**71**)	**0.0568**(**69**)
CCP	0.0491(88)	0.0492(83)	0.0486(67)	**0.0484**(**63**)	0.0488(76)
CH**	**0.1144**(**54**)$^{\S\Vert}$	**0.1144**(**61**)$^{\S\Vert}$	0.1145(62)$^{\S\Vert}$	0.1167(101)	0.1165(99)
CNS	0.0608(79)	0.0605(73)	**0.0603**(**71**)	0.0605(75)	0.0605(78)
Rank	5(63)	4(49)	3(48)	2(34)	**1(31)**

Table 7: RMSE for $K_{rel} = 40$.

A drawback of RS-K-medoids++, RS-UPGMA, and RS-maximin is that if the data contains anomalies, they are prone to selecting them as reference points. This limitation is probably why Random gets smaller RMSE than RS-UPGMA and RS-maximin for the CH data set with small-to-moderate K_{rel}, since that data set is known to contain some large anomalies. Therefore, we combined a simple anomaly detection method (k-nearest neighbors) with RS-UPGMA and tested it with the CH data set. We observed that anomaly detection improved the test error for RS-UPGMA ($K_{rel} = 5, 10, 20, 40$). Similar observations can also be drawn from the results for the S1 data set. S1 is the cleanest data set in our experiments: all input points are mapped to output points with sine-based function evaluations and without any distortions. Based on Tables 4–7, RS-UPGMA and RS-maximin have the largest error differences compared to Random for the S1 data set than any other data set. Therefore, a robust variant of the MLM, combined with RS-UPGMA or RS-maximin, should be considered for regression tasks with anomalies.

5.4 Case S1: Method Comparison

To demonstrate the differences among the five approaches we examined, we ran the reference point selection methods only for the S1 data, considering 100 reference points (10%). In Figure 2 (Appendix B), the smallest 500 pairwise Euclidean distances for the selected 100 reference points in the S1 data set are plotted in ascending order. Figure 2 also illustrates the differences between the reference point approaches. Overall, Random selection is the worst method and RS-maximin is the best method for identifying separate and input space, covering sets of reference points in a well-balanced manner. Interestingly, the ordering of the methods' pairwise distance curves is the same as the ordering of the methods' RMSE performance.

As noted in the results of Section 5.3, variances are not equal for several data sets, based on the Brown-Forsythe test. Clustering-based reference point selection gives smaller variances than the Random method for a small K_{rel}. In Appendix B, this difference is illustrated in Figure 3 and Figure 4 for the S1 data set. RMSE variance for the Random method is eight times larger compared to the RS-maximin method when $K_{rel} = 5$. When K_{rel} reaches 40, the variances are equal.

5.5 Discussion

We evaluated four clustering-based methods for reference point selection with the MLM. We focused on testing the methods against the Random approach in regression tasks with 15 data sets. An extensive experimental evaluation of the methods showed that the clustering-based methods can improve the MLM's performance. A good set of reference points is able to cover the data space well. When an optimal number of reference points is desired, RS-UPGMA and RS-maximin are valid choices. With respect to accuracy for a fixed number of reference points K, RS-maximin is the best choice for low K values ($K_{rel} = 5, 10$). For higher K values ($K_{rel} = 20, 40$), RS-UPGMA and RS-maximin are the best choices. However, RS-maximin is the most efficient approach since the computational cost with respect to the number of observations N is linear compared to RS-UPGMA, which has a quadratic time complexity with respect to N. Together with the LLS method for the second step of the MLM, we obtain, on the whole, a very computationally efficient approach. Note that it is enough to run deterministic reference point selection method once for each data set in hyperparameter tuning, while—for example—RS-K-medoids++ must be run for each hyperparameter value from the start. Moreover, the deterministic reference point selection methods reduce the MLM model's space and computational complexity because they can build the optimal model with smaller sets of reference points.

The conclusion by Pekalska et al. (2006) to favor a deterministic strategy agrees with our results on the quality of the deterministic RS-maximin. Even though the maximin method is not recommended to be used for the K-means initialization, based on the extensive study by Celebi et al. (2013), our study shows it to be a valid method for selecting reference points in the MLM. This difference highlights that reference point selection has a different aim than clustering or the initialization of a specific clustering method. For example, based on the performed experiments, the maximin method selects points such that extreme points are very valuable if they are not anomalies. Contrarily, in terms of K-means initialization, those points are far from the cluster centers. Hence, they are not optimal choices for clustering initialization.

Finally, the clustering-based methods are less robust for outliers than the Random approach. Therefore, an integration with outlier detection or the use of a robust approach for input and output distance matrix mapping should be considered for distorted data sets. Based on the experiments, reference point selection appears to control the balance between the regression model's interpolation and extrapolation. Selecting reference points from the data clouds' boundaries improves extrapolation abilities, but this approach might lead (in rare cases) to worse interpolation in the dense areas, as likely occurred for the CH data set.

6. Conclusion

In this paper, we addressed important open questions related to MLM research. Based on previous related works, we demonstrated the theory behind the MLM's interpolation and universal approximation properties by considering the behavior of its two main components: the linear mapping between distance matrices and the multilateration for output estimation. Our results ensure the MLM's generalization capability and indicate reference points' role in the bounded estimation error.

Motivated by our findings, we performed comprehensive computation experiments to evaluate different clustering-based approaches to reference point selection for the MLM in regression scenarios. In summary, all the methods performed better than standard random selection. The RS-maximin approach was the best choice due to its greater generalization capability, compact model size, simplicity, and more efficient computational implementation. In general, our experimental results demonstrate how the utilization of a heterogeneous pool of clustering methods, with respect to the characteristics of an unsupervised problem as part of a supervised method, can provide useful insights to the underlying problem.

In the future, adapting and evaluating the presented methods for classification tasks would also be interesting. Moreover, we could also analyze how such methods are affected when the number of reference points differs for input and output spaces. The latter consideration may modify the reference point selection problem, and it might result in additional interpretations of the MLM's generalization capability.

Acknowledgments

This work was supported by the Academy of Finland from grants 311877 and 315550. The authors would like to thank the Cearense Foundation for the Support of Scientific and Technological Development (FUNCAP) for its financial support.

Appendix A. Localization Linear System

Consider $\mathcal{Z}$, a set of known points in $\mathbb{R}^S$. Suppose the existence of $\boldsymbol{w} \in \mathbb{R}^S$ is unknown, but whose distances for each $\boldsymbol{z}_i \in \mathcal{Z}$, given by $||\boldsymbol{w} - \boldsymbol{z}_i||^2 = d_i^2$, are known. Suppose that we have another point $\boldsymbol{r} \in \mathbb{R}^S$, called benchmark-anchor-node (BAN), such that $||\boldsymbol{w} - \boldsymbol{r}||^2 = d_r^2$ and $||\boldsymbol{z}_i - \boldsymbol{r}||^2 = d_{ir}^2$ are also known. Thus, we have:

$$\begin{aligned}
d(\boldsymbol{z}_i, \boldsymbol{w})^2 =& ||\boldsymbol{w} - \boldsymbol{z}_i||^2 \\
d_i^2 =& \sum_{j=1}^{S} (w_j - z_{i,j})^2 \\
d_i^2 =& \sum_{j=1}^{S} (w_j - r_j + r_j - z_{i,j})^2 \\
d_i^2 =& \sum_{j=1}^{S} [(w_j - r_j) + (r_j - z_{i,j})]^2 \\
d_i^2 =& \sum_{j=1}^{S} [(w_j - r_j) - (z_{i,j} - r_j)]^2 \\
d_i^2 =& \sum_{j=1}^{S} [(w_j - r_j)^2 + (z_{i,j} - r_j)^2 - 2(w_j - r_j)(z_{i,j} - r_j)] \\
d_i^2 =& \underbrace{\sum_{j=1}^{S} (w_j - r_j)^2}_{d(\boldsymbol{w},\boldsymbol{r})^2} + \underbrace{\sum_{j=1}^{S} (z_{i,j} - r_j)^2}_{d(\boldsymbol{r},\boldsymbol{z}_i)^2} - 2\sum_{j=1}^{S} (w_j - r_j)(z_{i,j} - r_j) \\
d_i^2 - d_r^2 - d_{ir}^2 =& -2\sum_{j=1}^{S} (w_j - r_j)(z_{i,j} - r_j) \\
\sum_{j=1}^{S} \underbrace{(w_j - r_j)}_{\theta_j} \underbrace{(z_{i,j} - r_j)}_{A_{ij}} =& \underbrace{\frac{1}{2}[d_r^2 + d_{ir}^2 - d_i^2]}_{b_i} \\
\boldsymbol{A\theta} =& \boldsymbol{b} \qquad (9)
\end{aligned}$$

Thus, after solving the system $\boldsymbol{A\theta} = \boldsymbol{b}$, we compute $\boldsymbol{w} = \boldsymbol{\theta} + \boldsymbol{r}$ to recover the position of $\boldsymbol{w}$. Note that the BAN $\boldsymbol{r}$ can be selected from $\boldsymbol{Z}$ and thus satisfy all the necessary conditions for the application of the technique.

Appendix B. Figures

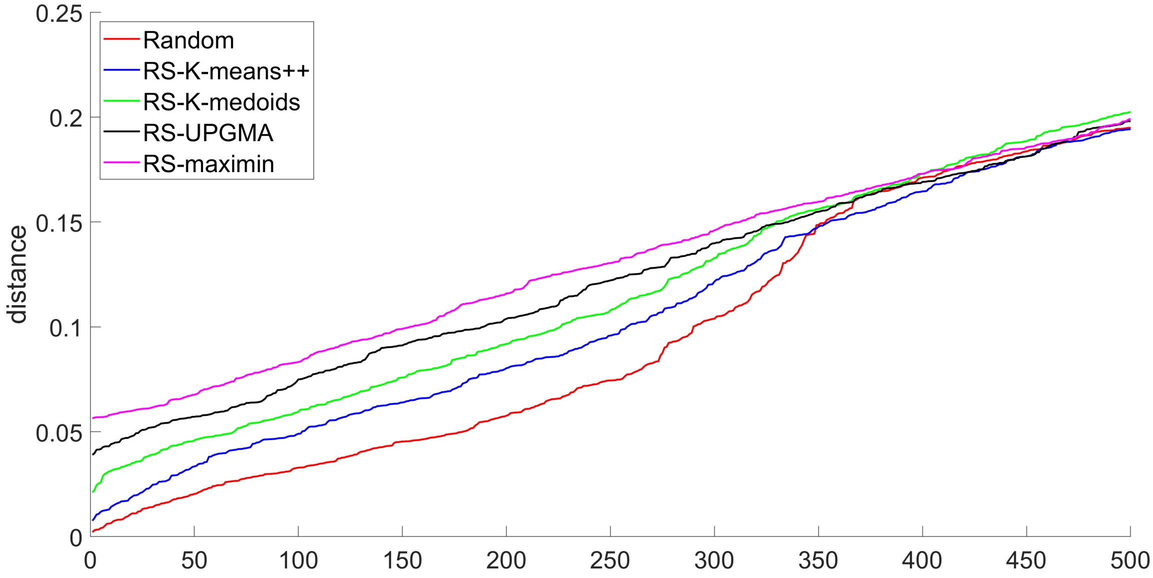

Figure 2: The smallest 500 pairwise Euclidean distances for the selected 100 reference points for S1 in ascending order. Clustering-based methods select a set of reference points that are more separeted each other compared to the random approach.

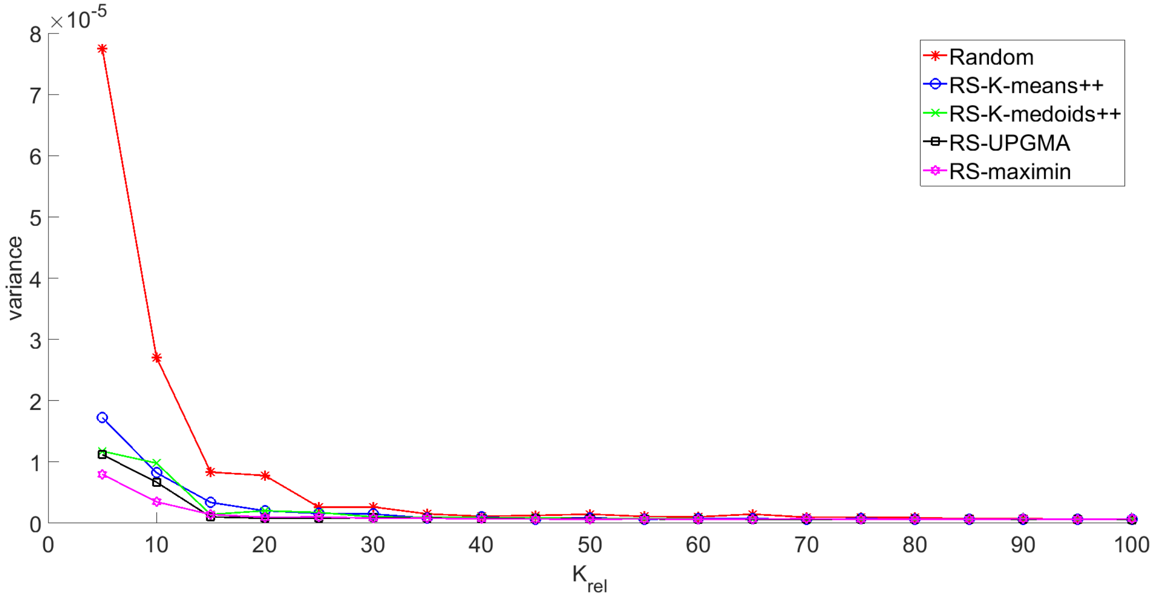

Figure 3: Variances of the RMSE test errors for the S1 data set.

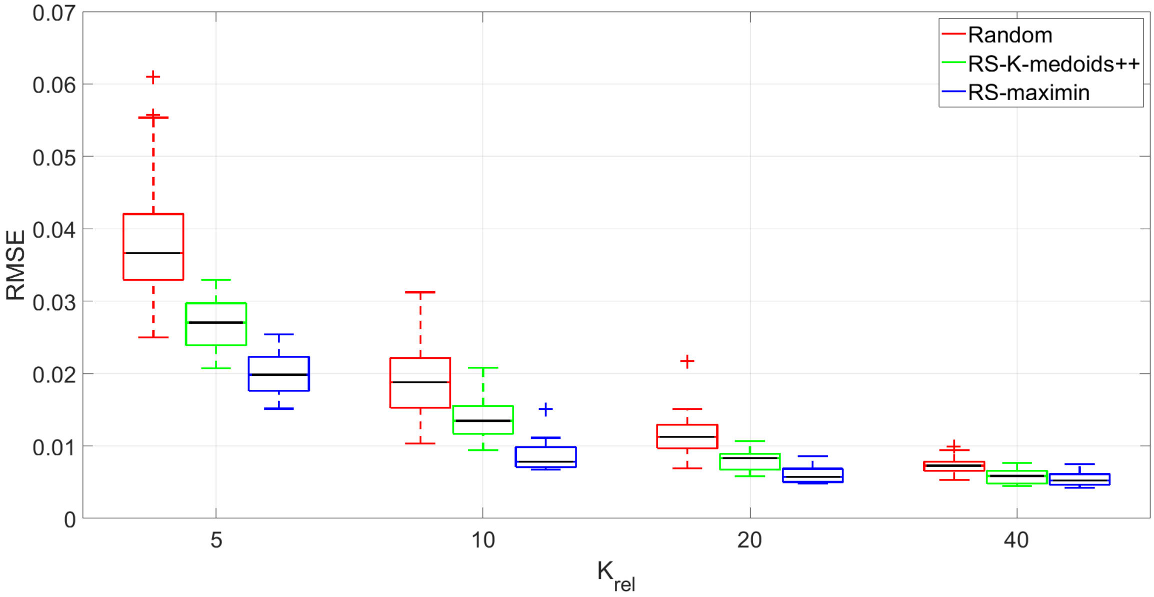

Figure 4: Boxplot of the RMSE test errors for the S1 data set.

References

Amaia Abanda, Usue Mori, and Jose A. Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, 2019.

Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and S. Yu Philip. Active learning: A survey. In *Data Classification*, pages 599–634. Chapman and Hall/CRC, 2014.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

John W. Auer. An elementary proof of the invertibility of distance matrices. *Linear and Multilinear Algebra*, 40(2):119–124, 1995.

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.

David S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

Weslley L. Caldas, João P. P. Gomes, and Diego P. P. Mesquita. Fast Co-MLM: An efficient semi-supervised Co-training method based on the minimal learning machine. *New Generation Computing*, 36(1):41–58, 2018.

Hongliu Cao, Simon Bernard, Robert Sabourin, and Laurent Heutte. Random forest dissimilarity based multi-view learning for radiomics application. *Pattern Recognition*, 88: 185–197, 2019.

M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

Yihua Chen. *Strategies for similarity-based learning*. PhD thesis, University of Washington, Program of Electrical Engineering, 2010.

Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776, 2009.

David N. Coelho, Guilherme A. Barreto, Cláudio M. S. Medeiros, and José D. A. Santos. Performance comparison of classifiers in the detection of short circuit incipient fault in a three-phase induction motor. In *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pages 42–48, 2014.

Yandre M. G. Costa, Diego Bertolini, Alceu S. Britto, George D. C. Cavalcanti, and Luiz E. S. Oliveira. The dissimilarity approach: a review. *Artificial Intelligence Review*, 53: 2783–2808, 2020.

Adonias C. de Oliveira, João P. P. Gomes, Ajalmar R. Rocha Neto, and Amauri H. de Souza Junior. Efficient minimal learning machines with reject option. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 397–402, 2016.

Amauri H. de Souza Junior, Francesco Corona, Yoan Miche, Amaury Lendasse, Guilherme A. Barreto, and Olli Simula. Minimal learning machine: A new distance-based method for supervised learning. In *International Work-Conference on Artificial Neural Networks*, pages 408–416. Springer, 2013.

Amauri H. de Souza Junior, Francesco Corona, Guilherme A. Barreto, Yoan Miche, and Amaury Lendasse. Minimal learning machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164:34–44, Sep 21 2015.

Madson L. D. Dias, Lucas S. de Souza, Ajalmar R. da Rocha Neto, and Amauri H. de Souza Junior. Opposite neighborhood: a new method to select reference points of minimal learning machines. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2018*, pages 201–206, 2018.

Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

José A. V. Florêncio, Madson L. D. Dias, Ajalmar R. da Rocha Neto, and Amauri H. de Souza Júnior. A fuzzy c-means-based approach for selecting reference points in minimal learning machines. In Guilherme A. Barreto and Ricardo Coelho, editors, *Fuzzy Information Processing*, pages 398–407, Cham, 2018. Springer International Publishing.

José A. V. Florêncio, Saulo A. F. Oliveira, João P. P. Gomes, and Ajalmar R. Rocha Neto. A new perspective for minimal learning machines: A lightweight approach. *Neurocomputing*, 2020.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 2. Springer series in statistics New York, 2001.

Haitao Gan, Nong Sang, Rui Huang, Xiaojun Tong, and Zhiping Dan. Using clustering analysis to improve semi-supervised classification. *Neurocomputing*, 101:290–298, 2013.

Joao P. P. Gomes, Diego P. P. Mesquita, Ananda Freire, Amauri H. Souza Júnior, and Tommi 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*, pages 383–388, 2017.

Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Ilan Gronau and Shlomo Moran. Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210, 2007.

Joonas Hämäläinen. *Improvements and applications of the elements of prototype-based clustering*, volume 43 of JYU dissertations. University of Jyväskylä, 2018.

Joonas Hämäläinen and Tommi Kärkkäinen. Initialization of big data clustering using distributionally balanced folding. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2016*, pages 587–592, 2016.

Joonas Hämäläinen and Tommi Kärkkäinen. Newton's method for minimal learning machine. In *Computational Sciences and Artificial Intelligence in Industry – New digital technologies for solving future societal and economical challenges*. Springer, 2020. (to appear).

Joonas Hämäläinen, Susanne Jauhiainen, and Tommi Kärkkäinen. Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 10(3):105, 2017.

Yanjun Hu, Lei Zhang, Li Gao, Xiaoping Ma, and Enjie Ding. Linear system construction of multilateration based on error propagation estimation. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):154, Jun 2016.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18:6869–6898, 2017.

Purushottam Kar and Prateek Jain. Similarity-based learning via data driven embeddings. In *Advances in neural information processing systems*, pages 1998–2006, 2011.

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

Brian Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5 (4):287–364, 2013.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *Journal of Machine Learning Research*, 13:981–1006, 2012.

Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on Information Theory*, 28(2):129–137, 1982.

Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

Átilla N. Maia, Madson L. D. Dias, João P. P. Gomes, and Ajalmar R. da Rocha Neto. Optimally selected minimal learning machine. In Hujun Yin, David Camacho, Paulo Novais, and Antonio J. Tallón-Ballesteros, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2018*, pages 670–678, Cham, 2018. Springer International Publishing.

Leandro B. Marinho, Amauri H. de Souza Junior, and Pedro P. Rebouças Filho. A new approach to human activity recognition using machine learning techniques. In *International Conference on Intelligent Systems Design and Applications*, pages 529–538. Springer, 2016.

Leandro B. Marinho, Jefferson S. Almeida, João W. M. Souza, Victor H. C. Albuquerque, and Pedro 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*, 72:1–17, 2017.

Leandro B. Marinho, Pedro P. Rebouças Filho, Jefferson S. Almeida, João W. M. Souza, Amauri H. de Souza Junior, and Victor H. C. de Albuquerque. A novel mobile robot localization approach based on classification with rejection option using computer vision. *Computers & Electrical Engineering*, 68:26–43, 2018.

Diego P. P. Mesquita, João P. P. Gomes, and Amauri H. de Souza Junior. A minimal learning machine for datasets with missing values. In *22nd International Conference on Neural Information Processing - ICONIP 2015*, pages 565–572, 2015.

Diego P. P. Mesquita, João P. P. Gomes, and Amauri H. de Souza Junior. Ensemble of efficient minimal learning machines for classification and regression. *Neural Processing Letters*, 46(3):751–766, 2017a.

Diego P. P. Mesquita, João P. P. Gomes, Amauri H. de Souza Junior, and Juvêncio S. Nobre. Euclidean distance estimation in incomplete datasets. *Neurocomputing*, 248:11–18, 2017b.

Charles A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

J. G. Moreno-Torres, J. A. Sáez, and 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(8):1304–1312, 2012.

Eugène-Patrice Ndong Nguéma and Guillaume Saint-Pierre. Model-based classification with dissimilarities: a maximum likelihood approach. *Pattern Analysis and Applications*, 11 (3-4):281–298, 2008.

Dino Oglic and Thomas Gärtner. Nyström method with kernel k-means++ samples as landmarks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2652–2660. JMLR.org, 2017.

Pavel Paclık and Robert P. W. Duin. Dissimilarity-based classification of spectra: computational issues. *Real-Time Imaging*, 9(4):237–244, 2003.

Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

Jooyoung Park and Irwin W. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5(2):305–316, 1993.

Elzbieta Pekalska and Robert P. W. Duin. Automatic pattern recognition by similarity representations. *Electronics Letters*, 37(3):159–160, 2001.

Elzbieta Pekalska and Robert P. W. Duin. Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(6):729–744, 2008.

Elzbieta Pekalska, Pavel Paclik, and Robert P. W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2:175–211, 2001.

Elzbieta Pekalska, Robert P. W. Duin, and Pavel Paclik. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208, 2006.

Antti Pihlajamäki, Joonas Hämäläinen, Joakim Linja, Paavo Nieminen, Sami Malola, Tommi Kärkkäinen, and Hannu Häkkinen. Monte Carlo simulations of Au38(SCH3)24 nanocluster using distance-based machine learning methods. *The Journal of Physical Chemistry A*, 2020.

Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Heydi Méndez-Vázquez, Edel García-Reyes, and Robert P. W. Duin. Towards scalable prototype selection by genetic algorithms with fast criteria. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 343–352. Springer, 2014.

Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Heydi Méndez-Vázquez, Edel García-Reyes, and Robert P. W. Duin. Scalable prototype selection by genetic algorithms and hashing. *arXiv preprint arXiv:1712.09277*, 2017.

Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

Farhad Pourkamali-Anaraki, Stephen Becker, and Michael B. Wakin. Randomized clustered Nystrom for large-scale kernel machines. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Michael J. D. Powell. Radial basis function for multivariable interpolation: a review. In *Algorithms for Approximation*, pages 143–167. Clarendon Press, Oxford, 1987.

Chandan K. Reddy and Bhanukiran Vinzamuri. A survey of partitional and hierarchical clustering algorithms. In *Data Clustering Algorithms and Applications*, pages 87–110. Chapman and Hall/CRC, 2013.

Frank-Michael Schleif and Peter Tino. Indefinite proximity learning: A review. *Neural Computation*, 27(10):2039–2096, 2015.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: from Theory to Algorithms.* Cambridge university press, 2014.

Robert R. Sokal. A statistical method for evaluating systematic relationship. *University of Kansas Science Bulletin*, 28:1409–1438, 1958.

Shiliang Sun, Jing Zhao, and Jiang Zhu. A review of Nyström methods for large-scale machine learning. *Information Fusion*, 26:36–48, 2015.

Liwei Wang, Cheng Yang, and Jufu Feng. On learning with dissimilarity functions. In *Proceedings of the 24th international conference on Machine learning*, pages 991–998, 2007.

Liwei Wang, Masashi Sugiyama, Cheng Yang, Kohei Hatano, and Jufu Feng. Theory and algorithm for learning with dissimilarity functions. *Neural computation*, 21(5):1459–1484, 2009.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.

P. Zerzucha, M. Daszykowski, and B. Walczak. Dissimilarity partial least squares applied to non-linear modeling problems. *Chemometrics and Intelligent Laboratory Systems*, 110 (1):156–162, 2012.

Kai Zhang and James T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, 21(10):1576–1587, 2010.

Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved Nyström low-rank approximation and error analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1232–1239, 2008.

Xueying Zhang, Qinbao Song, Guangtao Wang, Kaiyuan Zhang, Liang He, and Xiaolin Jia. A dissimilarity-based imbalance data classification algorithm. *Applied Intelligence*, 42(3): 544–565, 2015.