Newton's Method for Minimal Learning Machine

Joonas Hämäläinen and Tommi Kärkkäinen

Abstract Minimal Learning Machine (MLM) is a distance-based supervised machine learning method for classification and regression problems. Its main advances are simple formulation and fast learning. Computing the MLM prediction in regression requires solution of the optimization problem, which is determined by the input and output distance matrix mappings. In this paper, we propose to use the Newton's method for solving this optimization problem in multi-output regression and compare the performance of this algorithm with the most popular Levenberg-Marquardt method. According to our knowledge, MLM has not been previously studied in the context of multi-output regression in the literature. In addition, we propose new initialization methods to speed up the local search of the second-order methods.

1 Introduction

In multi-output regression, the aim is to predict multiple real-valued target variables all at once [2]. The most straightforward or baseline approach to build a multi-output regression model is to train an independent regression model for each target variable separately [12]. It has been experimentally demonstrated that using single model for the multi-output regression can give better prediction accuracy than the single target, vector-valued approach, especially when the target variables are correlated [12]. Other benefits of using multi-output over single target regression are lower computational cost, simpler models, and better model interpretability [2].

Joonas Hämäläinen
University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyvaskyla, Finland, e-mail: joonas.k.hamalainen@jyu.fi

Tommi Kärkkäinen
University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyvaskyla, Finland e-mail: tommi.karkkainen@jyu.fi

Minimal Learning Machine (MLM) [3] is a simple method with fast learning rate based on a linear mapping between distance matrices. These distance matrices are computed with respect to a subset of data points referred as reference points. According to [7, 6], favoring well-separed reference points improves the generalization capability of MLM and gives more compact models. Moreover, due to the construction of the distance-based kernel, MLM is capable to learn non-linear relations from data. Lately, it was shown that MLM has universal approximator capability and it can provide accurate predictions with the distance matrix regression model [6].

In the multilateration optimization problem of MLM, the predicted output is solved from the minimization problem determined by the predicted distances with respect to reference points. In one dimensional regression, the MLM prediction can be computed efficiently from the predicted distances with the analytical formula [11]. However, in the multi-output regression, such an approach can not be applied simultaneously to all target variables and we have to use an optimization solver to get the MLM prediction for the multidimensional output space. In the original formulation of the MLM [3], the Levenberg-Marquardt (LM) method [10] was recommended to be used to solve the multilateration problem. Another efficient method to solve such optimization problems is to apply the Newton's method [9].

In general, it is known that the second-order Newton's method converges to local minimum faster than the Levenberg-Marquadt method [4]. On the other hand, the Newton's method is more sensitive to the initialization than the Levenberg-Marquadt. Therefore, a good initial guess is essential for the Newton's method so that it can be applied with the MLM reliably. In this chapter, we integrate the Newton's method to MLM, propose efficient initialization methods for the Newton's method, and show experimental comparison with the Newton's and Levenberg-Marquadt methods for MLM in multi-output regression. In addition, we show a comparison of the multi-output MLM and single target MLM in multi-output regression.

2 MLM

2.1 Formulae

In general, training and prediction with MLM is straightforward. Training of the MLM method can be divided into three main steps:

1) Reference point selection,
2) Distance matrix computation,
3) Distance regression model computation.

The MLM training steps are depicted in Algorithm 1. In training, Step 3 is usually the most costly. Output predicting for new inputs with the MLM can also be divided into three steps:

1) Input space distance computation,

2) Output space distance estimation,
3) Optimization.

The MLM output prediction steps are depicted in Algorithm 2. In output prediction, various optimization methods can be used to solve the optimization problem which significantly affect how accurate and efficient MLM is for new inputs.

Algorithm 1 MLM training

Input: Training input dataset $\mathbf{X}$, Training output dataset $\mathbf{Y}$, #reference points K.
Output: Regression model $\mathbf{B}$, input space reference points $\mathbf{R}$, output space reference points $\mathbf{T}$.
1.1: Select set of input space reference points $\mathbf{R}$ from $\mathbf{X}$.
1.2: Select set of output space reference points $\mathbf{T}$ from $\mathbf{Y}$.
2.1: Compute distance matrix $\mathbf{D}_\mathbf{x}$ between $\mathbf{X}$ and $\mathbf{R}$.
2.2: Compute distance matrix $\mathbf{D}_\mathbf{y}$ between $\mathbf{Y}$ and $\mathbf{T}$.
3: Compute distance regression model $\mathbf{B}$ for $\mathbf{D}_\mathbf{x}$ and $\mathbf{D}_\mathbf{y}$ from (3).

Algorithm 2 MLM prediction

Input: New input $\mathbf{x}$, regression model $\mathbf{B}$, input space reference points $\mathbf{R}$, output space reference points $\mathbf{T}$.
Output: Predicted output $\mathbf{y}$.
1: Compute distances in input space $d_\mathbf{x} = (\|\mathbf{x} - \mathbf{r}_1\|, \ldots, \|\mathbf{x} - \mathbf{r}_K\|)$.
2: Estimate distances in output space $\boldsymbol{\delta} = d_\mathbf{x}\mathbf{B}$.
3: Solve multilateration problem (3) for $\mathbf{T}$ and $\boldsymbol{\delta}$.

Given set of input data points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and corresponding output data points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^M$ and $\mathbf{y}_i \in \mathbb{R}^L$, in the training phase of MLM, a set of rerence points $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^K$ is selected from $\mathbf{X}$ and $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^K$ from $\mathbf{Y}$. The number of reference points K could be also different for $\mathbf{T}$ and $\mathbf{R}$, however, they are usually selected equal. This simplifies the MLM formulation so that K is the only hyperparameter to be optimized in the training. In the original derivation of the MLM [3], the reference point selection was conducted randomly. For better generalization and more sparser models, the RS-maximin method for the reference point selection is preferred [7, 6]. Next, a distance matrix $\mathbf{D}_\mathbf{x} \in \mathbb{R}^{N\times K}$ for the input space and $\mathbf{D}_\mathbf{y} \in \mathbb{R}^{N\times K}$ for the output space is computed so that

$$(\mathbf{D}_\mathbf{x})_{ij} = \|\mathbf{x}_i - \mathbf{r}_j\|, \tag{1}$$

and

$$(\mathbf{D}_\mathbf{y})_{ij} = \|\mathbf{y}_i - \mathbf{t}_j\|. \tag{2}$$

Finally, a linear regression model between the distance matrices $\mathbf{D}_\mathbf{x}$ and $\mathbf{D}_\mathbf{y}$ is computed using the Ordinary Linear Least Squares (OLLS) [3]

$$\mathbf{B} = (\mathbf{D}_\mathbf{x}{}^\top\mathbf{D}_\mathbf{x})^{-1}\mathbf{D}_\mathbf{x}{}^\top\mathbf{D}_\mathbf{y}. \tag{3}$$

To assure positive definiteness and unique solvability (see, e.g., [8]), a small positive regularization constant $\varepsilon > 0$ times an identity matrix $\boldsymbol{I}$ can be added to the diagonal of the outer-product matrix to have it in the form $\mathbf{D_x}^\top \mathbf{D_x} + \varepsilon \boldsymbol{I}$ [7].

In the output prediction, we first compute distances in the input space for a new input $\mathbf{x}$ with respect to the reference points $\mathbf{R}$. In Step 2, the distance regression model $\mathbf{B}$ in Eq. (3) is applied to estimate the distances in the output space with respect to the reference points $\mathbf{T}$. Finally, to predict the output, we solve the multilateration problem, i.e., we aim to find $\mathbf{y}$ such that

$$\|\mathbf{y} - \mathbf{t}_k\| \approx \boldsymbol{\delta}_k, \tag{4}$$

where $k = 1, ..., K$ and $\boldsymbol{\delta} = (\|\mathbf{x} - \mathbf{r}_1\|, ..., \|\mathbf{x} - \mathbf{r}_K\|)\mathbf{B}$. This goal can be approximately achieved by solving the least-squares optimization problem, i.e., by minimizing the following objective function

$$J(\mathbf{y}) = \sum_{k=1}^{K} (\|\mathbf{y} - \mathbf{t}_k\|^2 - \boldsymbol{\delta}_k{}^2)^2. \tag{5}$$

With the one dimensional output space, we can differentiate the objective function (5) and set the resulting derivative to zero. This yields to the cubic equation [11]

$$Ky^3 - 3\sum_{i=1}^{K} t_k y^2 + \sum_{i=1}^{K} (3t_k^2 - \boldsymbol{\delta}_k) y + \sum_{i=1}^{K} (\boldsymbol{\delta}_k{}^2 t_k - t_k^3) = 0. \tag{6}$$

Solving the cubic equation with analytical formula is very fast compared to optimization based approaches to compute the MLM prediction for one dimensional regression. This MLM approach is referred as the Cubic Minimal Learning Machine (C-MLM) [11]. For a multi-output regression, C-MLM can be used to form separately regression models for each output variable. However, drawback of this single target MLM approach compared to forming single regression model for multi-output regression problem is that computational and space complexity for the MLM training is significantly higher compared to a single model approach. Therefore, optimization based solver for multi-outputs of the objective function (5), such as the Newton's method, is more preferred in terms of the computational and space complexity.

2.2 MLM using the Newton's method

In principle, the most efficient local solver for the quadratic (actually quartic) optimization problem is provided by the classic Newton's method with an iteration step$l \mapsto l + 1$ as follows:

$$\tilde{\boldsymbol{y}}^{l+1} = \tilde{\boldsymbol{y}}^{l} - \left[\nabla^2 \mathcal{J}(\tilde{\boldsymbol{y}}^{l})\right]^{-1} \nabla \mathcal{J}(\tilde{\boldsymbol{y}}^{l})$$

(e.g., [9] and articles therein). More precisely, it is straightforward to calculate the derivative and Hessian of Eq. (5) in a matrix-vector form. The most important facet for the efficiency of the Newton's method is the initial guess, which should be accurate enough to assure quadratic convergence, which is only obtained locally.

Here we suggest two simple initialization approaches for the multilateration problem. The first approach is based on considering Eqs. (4) and (5) in a completely component wise form

$$\sum_{k=1}^{K}\sum_{i=1}^{m}\left(\tilde{\boldsymbol{y}}^0-\boldsymbol{t}_k\right)_i^2\approx\sum_{k=1}^{K}\boldsymbol{\delta}_k^2=\sum_{k=1}^{K}\sum_{i=1}^{m}\frac{1}{m}\boldsymbol{\delta}_k^2,$$

which yields

$$\tilde{\boldsymbol{y}}_i^0=\frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{t}_k\pm\frac{\boldsymbol{\delta}_k}{\sqrt{m}}\right)_i. \tag{7}$$

In practice, all sign combinations for all components of $\tilde{\boldsymbol{y}}^0$ should be tested and the candidate with the smallest value of the cost function in Eq. (5) selected. We refer to this approach as delta±.

The second approach is based on using the K_t nearest reference points given by the prediction of the distance regression model. First, we identify K_t nearest reference points $\tilde{\mathbf{T}}=\{\tilde{\mathbf{t}}_k\}_{k=1}^{K_t}$ from $\mathbf{T}$ using predicted distances $\boldsymbol{\delta}$. Then, the initialization is simply given by

$$\tilde{\boldsymbol{y}}_i^0=\frac{1}{K_t}\sum_{k=1}^{K_t}\tilde{\mathbf{t}}_k. \tag{8}$$

For $K_t=1$, the initial guess is the nearest reference point based on the predicted distances. For $K_t=K$ it approaches the mean of data when K increases. Setting $1<K_t<<K$ we get an initial guess that is based on local neighborhood of the reference points. We refer to this approach as t-mean.

3 Experiments

Next, we describe the realization of the proposed methods and provide results from the computational experiments. The comparative starting point for the prediction accuracy experiments performed and reported here is the experimental conclusion that was drawn in [7] for the multi-output classification problems: The MLM does not overlearn so that the most accurate classifier was given by the parameter-free full (i.e., use all observations as reference points) MLM model.

Name	Description	N	M	L
ATP7D	Airline ticket minimum price over next 7 days for six flight cases.	296	411	6
ATP1D	Next day airline ticket price for six flight cases.	337	411	6
WQ	River water quality.	1060	16	14
SCM20D	Supply chain product mean price over next 20 days for 16 products.	8966	61	16
SCM1D	Next day supply chain product mean price for 16 products.	9803	280	16

Table 1: Dataset description and charasteristics.

3.1 Experimental setup

In the experiments, we first focus on comparing the Newton's method and the Levenberg-Marquardt for the MLM with different initialization methods. For fair comparison, we use same initialization methods for both. In addition to delta± and t-mean methods, we also used random reference point initialization to demonstrate the importance of proper initial guess. All the experiments are conducted in the MATLAB environment. For the Newton's method, we used our own matlab implementation and for the Levenberg-Marquardt the MATLAB's *lsqnonlin*-function with an option *options.Algorithm* = *'levenberg-marquardt'*. For the both method's we used 10^{-4} relative step tolerance as the stopping criteria[1] and maximum number of iterations to 400. Since the Levenberg-Marquardt build-in implementation also uses the function tolerance[1], we set to this to 10^{-15} so that it will not be used as a stopping criteria. In addition, we show comparison of the Newton's method to single target C-MLM. The C-MLM output prediction was implemented based on the Algorithm 2 in [11].

We selected five multi-output regression datasets that have real-valued target vectors from `http://mulan.sourceforge.net/datasets-mtr.html`. The characteristics of the datasets are listed in Table 1, where N is the number of observations, M is the number of input variables and L is the number of target variables. Detailed description of the Airline Ticket Price (ATP) and the Supply Chain Management (SCM) datasets is given in [13].

The ATP1D [13] dataset is a high-dimensional regression dataset for the next day flight ticket prices prediction and, similarly, the ATP7D dataset is for the minimum ticket prices prediction for the next 7 days, i.e., the whole week forward. In the ATP datasets, these statistics are computed for the same six target flight cases. Note that ATP1D and ATPD7 have the same input variables. The input variables for the ATP datasets contain information about the number of days between observation day and the departure day of the flight, day of the week, multiple variables related to the price statics and the quotes from all airlines.

The Water Quality (WQ) dataset consist of 14 target variables that are collected from chemical analysis of the water quality of Slovenian rivers during a six-year period [5]. The goal of the WQ dataset is to predict 14 chemical chemical variables

[1] `https://www.mathworks.com/help/optim/ug/tolerances-and-stopping-criteria.html`

with the 16 biological variables which describe the living organisms (biota) of the river.

The SCM datasets SCM1D and SCM20D consist of regression task for 16 output variables (product prices) which are next day mean prices (SCM1D) or 20 days mean prices (SCM20D). These datasets are collected from the Trading Agent Competition in SCM, where observation corresponds to single day in the competition. Input variables are related to observed prices on an observation day and time-delayd days.

We reduced the dimensions of the ATP1D and ATP7D input datasets by using the binary random projection [1] resulting to $M = 50$ dimensional input data. This way we can avoid distance matrix singularity issues, because originally the input dimension was very high and the number of data points relatively low. Use of the dimension reduction also decreases the computational load in distance calculations. Input and output datasets were then minmax-scaled to the range of $[0, 1]$.

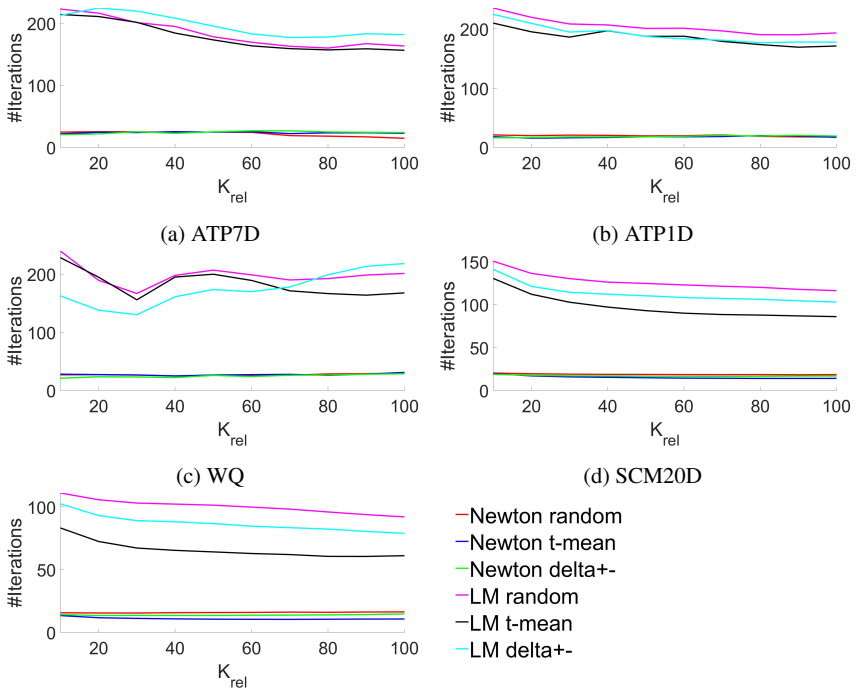
In each trial, we used a random partition of the whole dataset into training ($\frac{2}{3}$) and testing sets ($\frac{1}{3}$). We repeated these trials for each method ten times and averaged the results. For the prediction accuracy, we used the Root Mean Squared Error (RMSE) and computational cost of the optimization method was compared by tracking the number of iterations needed for the convergence. The RS-maximin method [6] was used for the reference point selection and the relative number of reference points $K_{rel} = 100\frac{K}{N_{train}}$ was varied as $\{10, 20, ..., 100\}$. For the delta±, we selected the initial point corresponding to the smallest objective function (5) value out of two sign combinations: $\mathbf{y}^0$ with only using the plus-sign or the minus sign in equation (7). For t-mean we fixed $K_t = 5$.

Let us summarize the computational complexity related to the experiments. Clearly, higher number of observations N, higher input dimension M, and higher-dimensional output L (vai C) increase the computational burden within the methods. With respect to these parameters, the increase is linear because these parameters only affect the computation of the distance matrices D_x and D_y in (1) and (2), respectively. The core parameter affecting the overall computational complexity of the methods is K (number of reference points). Computationally the most demanding operation is to solve the linear distance regression equation in (3), since, e.g., basic form of a Cholesky factorization needs $O(K^3)$ operations. Because of these considerations, the scalability of the methods concerning the parameters defined in Table 1 will be studied with respect to K only.

3.2 Results

3.2.1 Computational cost

The #iterations are plotted as a function of K_{rel} in Figure 1. There is almost an order of magnitude difference between the Newton's and LM method, the Newton's method clearly performs better than the LM method in terms of #iterations. The execution time (wall-clock and CPU time) of the LM method was even two orders

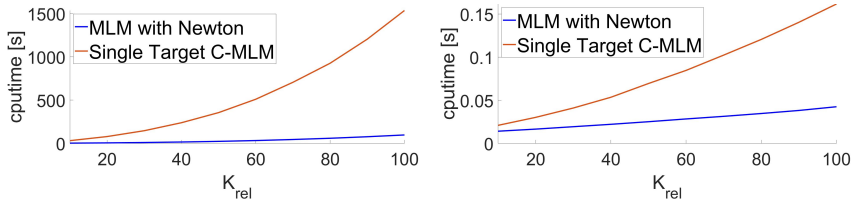


(a) ATP7D (b) ATP1D

(c) WQ (d) SCM20D

(e) SCM1D

Fig. 1: #Iterations as a function of K_{rel} for the Newton's and LM method.



(a) train (b) prediction

Fig. 2: CPU time as a function of K_{rel} for single target C-MLM and MLM using the Newton's method.

of magnitude slower than the Newton's method, this probably due to MATLAB's build-in implementation's high initial computational cost of the wrapper functions. Therefore, #iterations give more realistic comparison of the execution time between these methods.

The random approach for the initialization performs clearly worst for both optimization methods. For the two largest datasets, SCM1D and SCM20D, both methods convergence fastest with the t-mean initialization. Performance of the delta± com-

pared to other methods varies for different datasets, but in overal, also that method gives faster convergence for both methods than the random approach.

In addition, we compared the execution time for the single target C-MLM and the MLM with the Newton's method using the t-mean initialization. In general, training time of the single target C-MLM is clearly higher than the MLM with the Newton's method and the prediction time with the single target C-MLM is about equal or worse. Note that the initialization time of the Newton's method is also included in the prediction time. Results for the largest dataset are shown in Fig. 2. For the output prediction, the time difference increases when K increases.

3.2.2 Accuracy

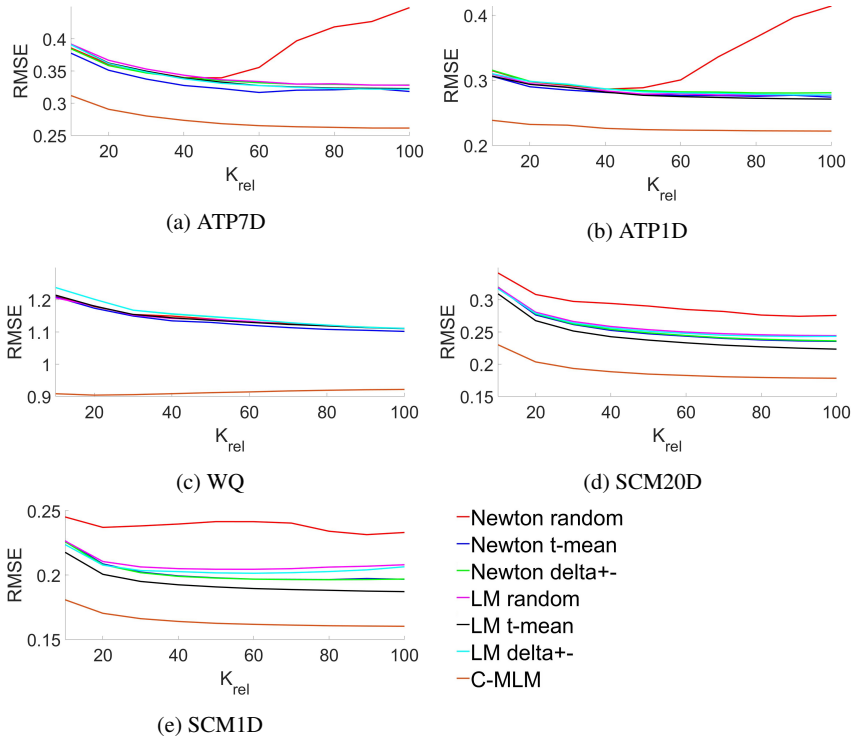
Results for the prediction accuracy are summarized in Fig. 3. The single target C-MLM is the most accurate method for all the datasets. Even for the $K_{rel} = 10$ it is more accurate than the full MLM model ($K_{rel} = 100$) with both optimization methods. As expected, the random initialization approach leads to worse RMSE results for the Newton's method but also slightly for the LM method, due to possible convergence to a bad local minimum. The delta± and t-mean initialization seem to give similar level of the RMSE accuracy for the prediction.

4 Discussion

The proposed initialization approaches for the optimization based solvers of the MLM's second phase objective function (5) can significantly affect to the generalization accuracy and convergence rate of the whole MLM method. Especially with the Newton's method, a proper initial guess needs to be used. The LM method is more robust to a bad initialization than the Newton's method, however, a good initialization also improves the convergence rate of LM. Using the Newton's method for the MLM output prediction can provide performance benefits over LM if the initial guess is close enough to the global solution.

However, because there are some differences in the generalization accuracy between the different initialization methods, we suggest generating multiple different initialization candidates provided by the proposed methods and then select the initial solution corresponding to the smallest objective function value. For delta± we can achieve this by testing multiple different sign combinations and for t-mean we can vary K_t variable. This could be alternated based on the demands of the application. If the number of reference points is tuned, e.g., with cross-validation based approach, generating just one initialization candidate is probably enough to capture the optimal K value.

Based on the results for the five real-valued multi-output regression datasets we ended up with the similar observation than in [7]: the MLM over-learning does not seem to occur! This implies that the reference point selection is more an issue to



(a) ATP7D (b) ATP1D

(c) WQ (d) SCM20D

(e) SCM1D

Fig. 3: MLM Prediction accuracy as a function of K_{rel}.

balance the computational cost and accuracy rather than to fine-tune the model for the optimal accuracy in the context of multi-output regression.

Based on the results, computational costs for the training and prediction are clearly higher in the multi-output regression for the single target C-MLM than when using the single model based MLM approach. On the other hand, using single target C-MLM regression models improves the accuracy compared to the single model approach. Therefore, the single target C-MLM should be considered for multi-output regression rather than the single MLM model if the computational cost remains reasonable and on the acceptable level.

5 Conclusions

In this chapter, we proposed and tested the Newton's method for the MLM output estimation in the context of multi-output regression. The MLM output estimation in regression requires to solve an optimization problem, where the algorithm details have significant effect to the accuracy and computational cost of the MLM.

Based on the experiments with five multi-output regression datasets, forming the MLM multi-output model with the single target MLM models is more accurate approach than using a single MLM model for all the targets. However, the computational cost for the MLM training and prediction can be clearly reduced when the regression problem has several targets by using the Newton's method with some loss in the generalization accuracy. How significant this loss is depends on the application.

The Newton's method convergences up to an order of magnitude faster than the Levenberg-Marquadt method for the MLM optimization problem with solving the MLM optimization equally well when the proper initial guess is given. We recommend selecting initial point corresponding to the smallest MLM objective function value out of multiple different initialization approaches for the Newton's method. In the future work, a possibility integrate/modify algorithms from the state-of-the-art multi-output regression approaches to the MLM should be studied.

Acknowledgements The authors would like to thank the Academy of Finland for the financial support (grants 311877 and 315550).

References

1. D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671 – 687, 2003, special Issue on {PODS} 2001.
2. H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.
3. A. H. de Souza Junior, F. Corona, G. A. Barreto, Y. Miche, and A. Lendasse, "Minimal learning machine: A novel supervised distance-based approach for regression and classification," *Neurocomputing*, vol. 164, pp. 34–44, 2015.
4. J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*. Siam, 1996, vol. 16.
5. S. Džeroski, D. Demšar, and J. Grbović, "Predicting chemical parameters of river water quality from bioindicator data," *Applied Intelligence*, vol. 13, no. 1, pp. 7–17, 2000.
6. J. Hämäläinen, A. S. Alencar, T. Kärkkäinen, C. L. Mattos, A. H. S. Júnior, and J. P. Gomes, "Minimal learning machine: Theoretical results and clustering-based reference point selection," *arXiv preprint arXiv:1909.09978*, 2019.
7. T. Kärkkäinen, "Extreme minimal learning machine: Ridge regression with distance-based basis," *Neurocomputing*, vol. 342, pp. 33–48, 2019.
8. T. Kärkkäinen and M. Saarela, "Robust principal component analysis of data with missing values," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2015, pp. 140–154.
9. C. T. Kelley, *Solving nonlinear equations with Newton's method*. Siam, 2003, vol. 1.
10. D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
11. D. P. P. Mesquita, J. P. P. Gomes, and A. H. Souza Junior, "Ensemble of efficient minimal learning machines for classification and regression," *Neural Processing Letters*, pp. 1–16, 2017.
12. E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-label classification methods for multi-target regression," *arXiv preprint arXiv:1211.6581*, pp. 1159–1168, 2012.
13. ——, "Multi-target regression via input space expansion: treating targets as inputs," *Machine Learning*, vol. 104, no. 1, pp. 55–98, 2016.