Instance-Based Multi-Label Classification via Multi-Target Distance Regression

Joonas Hämäläinen, Paavo Nieminen, Tommi Kärkkäinen *

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

Abstract. Interest in multi-target regression and multi-label classification techniques and their applications have been increasing lately. Here, we use the distance-based supervised method, minimal learning machine (MLM), as a base model for multi-label classification. We also propose and test a hybridization of unsupervised and supervised techniques, where prototype-based clustering is used to reduce both the training time and the overall model complexity. In computational experiments, competitive or improved quality of the obtained models compared to the state-of-the-art techniques was observed.

1 Introduction

Applications of supervised learning, where models are constructed to predict multiple target variables at once, rapidly increase their popularity. This research field within machine learning is referred as Multi-Output Learning [1], which can be divided into two main categories: *i*) Multi-Label Classification (MLC); and *ii*) Multi-Target Regression (MTR). In MLC, an instance is associated with multiple labels contrary to the conventional Single-Label Classification (SLC), where a single label is determined. There exists a plethora of methods for MLC, which can be divided into two main groups [2]: *i*) algorithm adaptation; and *ii*) problem transformation. In general, the distinction is made based on whether the classifier or the MLC problem itself is being modified. In algorithm adaptation, a specific classification method is tailored so it can be applied to MLC directly. The problem transformation methods modify the multi-label problem to be suitable for any single-label classifier.

A supervised distance-based method, the Minimal Learning Machine (MLM) [3], has been shown a promising performance in many experiments [4, 5, 6, 7]. Lately, MLM and the Extreme MLM (EMLM) [7], were identified to have appealing characteristics for MTR with problem transformation [8]. It has been demonstrated that the MLM avoids over-fitting for high-dimensional input spaces in classification [7] and regression [9, 4]. Therefore, tuning the MLM's only hyperparameter, the number of reference points, is mostly an issue of balancing the model complexity and the generalization capability in a straightforward manner: increasing the model complexity (the number of reference points) increases the generalization accuracy. However, increasing the accuracy of the model in this way comes with a cost, since the computational complexity of the training

*The work has been supported by the Academy of Finland from the projects 311877 and 315550.

phase behaves quadratically with respect to the number of reference points and linearly with respect to the number of observations [3].

In [10], it was shown that clustering application to MLC can be useful, especially for a large number of labels. In terms of high-dimensional output spaces, large-scale MLC problem arise from the application domains such as of image annotation [10] and text categorization [11]. In this paper, our aim is to adapt MLM to MLC and reduce the complexity of training and resulting models using clustering of input data. First, a straightforward Multi-Label MLM formulation is introduced based on the Nearest Neighbour MLM (NN-MLM) [6]. Then, the Clustering-Based ML-MLM (CBML-MLM) is proposed by utilizing the prototype-based clustering [12, 13]. Note that this technique readily supports federated learning scenarios [14].

2 Multi-Label Minimal Learning Machines

Suppose we have N input data points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^M$, and the corresponding 1-of-L encoded output vectors $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N,\ \mathbf{y}_i \in \{0,1\}^L$. Suppose we have selected a subset of the so-called reference points $\mathbf{R} = \{\mathbf{r}_k\}_{k=1}^K$ from $\mathbf{X}$, and the corresponding subset of points $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^K$ from $\mathbf{Y}$. In the MLM, the main idea is to learn a linear regression between the distance matrices $\mathbf{D_x} \in \mathbb{R}^{N\times K}$ and $\mathbf{D_y} \in \mathbb{R}^{N\times K}$, where $\mathbf{D}_{\mathbf{x}(i,j)} = d(\mathbf{x}_i, \mathbf{r}_j)$ and $\mathbf{D}_{\mathbf{y}(i,j)} = d(\mathbf{y}_i, \mathbf{t}_j)$ for the Euclidean distance $d(\cdot,\cdot)$. The Multi-Target Distance Regression (MTDR) model is then formed by utilizing the Ordinary Least Squares (OLS) ([3])

$$\mathbf{B} = ({\mathbf{D_x}}^T\mathbf{D_x})^{-1}{\mathbf{D_x}}^T\mathbf{D_y}, \tag{1}$$

where the $K\times K$ matrix $\mathbf{B}$ contains the coefficient for the MTDR model. As can seen from (1), the distance-based approach is especially beneficial for problems with high feature spaces and large number of classes, because M and L only affect construction of distance matrices but not the complexity of learning. Therefore, in off-the-shelf scenario, the distance-based regression can be more efficient and generalize better than deep learning models [5].

In the MLM prediction, for a given input $\mathbf{x}$, distances to the reference points $\mathbf{R}$ are computed and the MTDR model is used to predict the output space distances $\tilde{\boldsymbol{\delta}} = (\delta_1, ..., \delta_K)^T$ to the reference points $\mathbf{T}$. In [6], it was proved for SLC that assigning class label of the nearest output space reference point as a prediction $\mathbf{y}$, so that $\mathbf{y} = \mathbf{t}_q$ where $q = \text{argmin}_k\, \tilde{\boldsymbol{\delta}}_{(k)}$, is an optimal solution to the multilateration problem [3]. Note that solving the multilateration problem for the MLC problems with testing all the label combinations would be very complex and time-consuming, because the number of different label combinations could be huge for hundreds or even thousands of labels.

In here, we extent the NN-MLM approach straightforwardly to MLC so that the predicted set of labels is assigned directly from a set of labels associated with the nearest predicted output space reference point. We assume that the set of labels associated with the predicted nearest neighbour is a reasonable approximate solution to the multilateration problem. Furthermore, because this kind

Algorithm 1: CBML-MLM training

Input: Input data $\mathbf{X}$, output labels $\mathbf{Y}$, #clusters K_c, #clusters for distance regression fit $\tilde{K}$, prototype-based clustering algorithm f_c.

Output: Set of regression models $\{\mathbf{B_k}\}_{k=1}^{K_c}$, cluster-wise input space reference points $\{\mathbf{R}_k\}_{k=1}^{K_c}$, cluster-wise output space reference points $\{\mathbf{T}_k\}_{k=1}^{K_c}$, cluster prototypes $\{\mathbf{c}_k\}_{k=1}^{K_c}$.

$\{I_k\}_{k=1}^{K_c}, \{\mathbf{c}_k\}_{k=1}^{K_c} \leftarrow f_c(\mathbf{X}, K_c)$ // Indices I_k refer to clustering partition k
for $k \in \{1, \ldots, K_c\}$ **do**
  $\mathbf{R}_k, \mathbf{T}_k \leftarrow$ select subsets from $\mathbf{X}$ and $\mathbf{Y}$ according to I_k
  $\tilde{I}_k \leftarrow$ for c_k, find $\tilde{K}$ closest prototypes from $\{\mathbf{c}_k\}_{k=1}^{K_c}$
  $\mathbf{D_x}_k, \mathbf{D_y}_k \leftarrow$ compute cluster-wise distance matrices for $\{\mathbf{x}_i \mid i \in \bigcup_{k \in \tilde{I}_k} I_k\}$ and $\mathbf{R}_k$, and for, $\{\mathbf{y}_i \mid i \in \bigcup_{k \in \tilde{I}_k} I_k\}$ and $\mathbf{T}_k$
  $\mathbf{B}_k \leftarrow$ solve Eq. (1) for $\mathbf{D_x}_k$ and $\mathbf{D_y}_k$

of an approach relies on assigning a set of labels to an instance directly from a reference point, use of full MLM would ensure that all the possible label combinations occured in the training data are contained within possible predictions.

We will refer to this direct MLC algorithmic adaption as Multi-Label MLM (ML-MLM). In the categorization of MLC algorithms, the ML-MLM approach can be referred as an instance-based multi-label classifier similar to the ML-kNN method [15], where predicted set of labels is computed with the Maximum A Posteriori (MAP) method from the sets of labels related to the k nearest neighbours in the input space. ML-MLM identifies the nearest neighbour via the distance regression model while ML-kNN uses directly the input space.

Since ML-MLM selects all the data points as reference points, computational complexity of ML-MLM's training is $\mathcal{O}(N^3)$. To improve this high training cost, we propose a novel Clustering-Based ML-MLM (CBML-MLM) approach with reduced time complexity. However, we still will utilize all the data points as reference points to again ensure that the full diversity of the label combinations is preserved. The training algorithm for the proposed method is given in Algorithm 1 and the prediction phase is given in Algorithm 2. The training requires a prototype-based clustering algorithm f_c for partitioning the input space to local subsets. Prototype-based clustering methods such as K-means++ and K-spatialmedians++ [12] could be used. Both of these methods have linear time complexities and can be implemented in parallel for large-scale data sets [16, 13].

In the training phase, K_c local MTDR models are trained where each cluster's points are selected as reference points. For each local MTDR model, training data is formed from the union of data points belonging to the nearest $\tilde{K}$ clusters. For $\tilde{K} = 1$, the MTDR training data is the same as the local set of reference points, and for $\tilde{K} = K_c$, the whole data is utilized as training data. Note that the size of the final model is independent of parameter $\tilde{K}$. Similar to ML-MLM, CBML-MLM spends most of the training time solving the OLS from Eq. (1). For $\tilde{K} = 1$, the time complexity for training each cluster-wise model is

Algorithm 2: CBML-MLM prediction

Input: Input $\mathbf{x}$, a set regression models $\{\mathbf{B_k}\}_{k=1}^{K_c}$, cluster-wise input space reference points $\{\mathbf{R}_k\}_{k=1}^{K_c}$, cluster-wise output space reference points $\{\mathbf{T}_k\}_{k=1}^{K_c}$, cluster prototypes $\{\mathbf{c}_k\}_{k=1}^{K_c}$.

Output: Set of labels $\mathbf{y}$.

1: $k^* \leftarrow \text{argmin}_k\, d(\mathbf{x}, \mathbf{c}_k)$ // identify nearest prototype

2: $\mathbf{d_x} \leftarrow [d(\mathbf{x}, \mathbf{R}_{k^*(1)}), ..., d(\mathbf{x}, \mathbf{R}_{k^*(N_{k^*})})]$, where $N_{k^*} = |\mathbf{R}_{k^*}|$

3: $\tilde{\boldsymbol{\delta}} \leftarrow \mathbf{d_x}\mathbf{B}_{k^*}$ // predict distances with a local regression model

4: $q \leftarrow \text{argmin}_k\, \tilde{\boldsymbol{\delta}}_{(k)}$ // identify nearest neighbour with predicted distances

5: $\mathbf{y} \leftarrow \mathbf{T}_{k^*(q)}$.

$\mathcal{O}(N_k^3)$, where N_k is the number of observations in a cluster k. Therefore, the time complexity is $\mathcal{O}(N_*^3)$, where N_* denotes the number of observations in the largest cluster. For the other extreme, $\tilde{K} = K_c$, the cluster-wise training time complexity is $\mathcal{O}(N_k^2 N)$ which implies that the overall training time complexity is $\mathcal{O}(N_*^2 N)$. Note that if $N_* << N$, the CBML-MLM with $\tilde{K} = 1$ is clearly faster to train than the CBML-MLM with $\tilde{K} = K_c$. Moreover, if we have $N_* << N$, CBML-MLM is significantly faster to train than ML-MLM. In the prediction phase, the cluster prototypes are used for selecting the local MTDR model for classification.

3 Results

We selected six MLC data sets from `http://mulan.sourceforge.net` and utilized given training and testing data set division in order to be able compare our results to the results given in [17]. For the selected data sets, number of observations varied from 593 to 43907, number of features varied from 72 to 1001, number of labels varied from 6 to 374, and label cardinality varied from 1.1 to 4.4. We scaled all the input features to the range of $[0, 1]$. We selected ML-kNN as a main baseline, and fixed $k = 10$, similar to many other works [10]. For CBML-MLM, we used K-spatialmedians++ [12] with 100 repetitions as a clustering method, and selected $K_c = 10$ and $\tilde{K} = \{1, 10\}$. We did not perform any hyper-parameter optimization for CBML-MLM. In the experiments, the largest cluster size normalized by the number of training observations varied from 0.13 to 0.26. We used the existing MATLAB implementation of the ML-kNN [15] given in `http://www.lamda.nju.edu.cn/`. The proposed methods were implemented with MATLAB as well. For the evaluation of the classifiers' performance, we used two uncorrelated and recommended measures from [18]: hamming loss and accuracy (or example-based accuracy).

In Table 1, results for the experimented methods are shown in columns two to five. Moreover, in [17], Random Forest of Predictive Clustering Trees (RF-PCT) was the best performing method in the comparison. The results regarding RF-PCT for hamming loss and accuracy are shown in the last column. The best

Data set	ML-kNN	ML-MLM	CBML-MLM $\tilde{K}=1$	CBML-MLM $\tilde{K}=10$	Best from [17]
Emotions	0.21/0.51	0.20/**0.57**	0.21/**0.57**	0.21/**0.57**	**0.19**/0.52
Scene	0.10/0.66	**0.08**/**0.77**	0.10/0.72	0.09/0.74	0.09/0.54
Yeast	**0.20**/0.51	**0.20**/**0.55**	0.21/0.52	**0.20**/**0.55**	**0.20**/0.48
Enron	0.05/0.25	**0.03**/0.40	0.04/0.39	0.04/0.39	0.05/**0.42**
Corel5k	**0.01**/0.02	**0.01**/**0.18**	**0.01**/0.16	**0.01**/0.15	**0.01**/0.01
Mediamill	**0.03**/0.42	**0.03**/**0.48**	**0.03**/0.47	**0.03**/**0.48**	**0.03**/0.44
	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(N_*^3)$	$\mathcal{O}(N_*^2 N)$	$\mathcal{O}(Nlog(N))$

Table 1: Results for the hamming loss (hl) and accuracy metrics (acc). The elements in the table are formatted as hl/acc. For hl, a smaller value is better, for acc, a larger value is better. In the last row, the training time complexities are shown with respect to the data size N. The number of observations in the largest cluster is denoted as N_*.

results are emphasized in bold for each data set. The training time complexities of RF-PCT and ML-kNN are given in [19]. In Table 1, these are represented with respect to the data size. In terms of the evaluated metrics, CBML-MLM and ML-MLM methods clearly outperform the ML-kNN baseline. Moreover, compared to the best performing method in [17], CBML-MLM and ML-MLM have better accuracy than RF-PCT for four data sets, and for the Scene and Corel5k data sets, the accuracy difference is significant. In terms of hamming loss, the CBML-MLM and ML-MLM have similar performance to RF-PCT. This means that in particular CBML-MLM with $\tilde{K}=1$ provides learning efficiency, locality of models and therefore natural data parallelism, and high accuracy. Increasing size of the training data for the local MTDR models with the choice $\tilde{K}=K_c$ seems, only in some cases, slightly improve the CBML-MLM performance.

4 Conclusions

In this paper, we adapted and tested the minimal learning machine (MLM) in multi-label classification (MLC) problems for the first time. Experimental results showed that a state-of-the-art performance in MLC was reached with the proposed techniques. We adapted the nearest neighbor MLM (NN-MLM) approach to MLC, because in this way, the label correlations can be taken into account. Moreover, we showed that clustering can be applied to reduce the MLM's training time and model complexity with only a small sacrifice in accuracy and hamming loss. For the largest data set, this sacrifice was smallest which suggests that the proposed clustering-based MLM approach would be especially suited for large-scale MLC problems. As future work, we aim to cover, both methodologically and experimentally, the full scope of problem transformations in multi-target regression and classification problems using distance-based machine learning techniques.

References

[1] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE transactions on NNLS*, vol. 31, no. 7, pp. 2409–2429, 2019.

[2] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[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] A. Pihlajamäki, J. Hämäläinen, J. Linja, P. Nieminen, S. Malola, T. Kärkkäinen, and H. Häkkinen, "Monte carlo simulations of Au38(SCH3)24 nanocluster using distance-based machine learning methods," *The Journal of Physical Chemistry A*, vol. 124, no. 23, pp. 4827–4836, 2020.

[5] J. Linja, J. Hämäläinen, P. Nieminen, and T. Kärkkäinen, "Do randomized algorithms improve the efficiency of minimal learning machine?," *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, pp. 533–557, 2020.

[6] 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.

[7] T. Kärkkäinen, "Extreme minimal learning machine: Ridge regression with distance-based basis," *Neurocomputing*, vol. 342, pp. 33–48, 2019.

[8] J. Hämäläinen and T. Kärkkäinen, "Problem transformation methods with distance-based learning for multi-target regression," *ESANN*, 2020.

[9] J. Hämäläinen, A. S. Alencar, T. Kärkkäinen, C. L. Mattos, A. H. Souza Júnior, and J. P. Gomes, "Minimal learning machine: Theoretical results and clustering-based reference point selection," *Journal of Machine Learning Research*, vol. 21, 2020.

[10] G. Nasierding, G. Tsoumakas, and A. Z. Kouzani, "Clustering based multi-label classification for image annotation and retrieval," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 4514–4519, IEEE, 2009.

[11] M. Jiang, Z. Pan, and N. Li, "Multi-label text categorization using l21-norm minimization extreme learning machine," *Neurocomputing*, vol. 261, pp. 4–10, 2017.

[12] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen, "Comparison of internal clustering validation indices for prototype-based clustering," *Algorithms*, vol. 10, no. 3, p. 105, 2017.

[13] J. Hämäläinen, T. Kärkkäinen, and T. Rossi, "Scalable robust clustering method for large and sparse data," *ESANN*, 2018.

[14] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys & Tutorials*, pp. 1–49, 2021.

[15] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[16] J. Hämäläinen, T. Kärkkäinen, and T. Rossi, "Improving scalable k-means++," *Algorithms*, vol. 14, no. 1, p. 6, 2021.

[17] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[18] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann, "Correlation analysis of performance measures for multi-label classification," *Information Processing & Management*, vol. 54, no. 3, pp. 359–369, 2018.

[19] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, "Comprehensive comparative study of multi-label classification methods," *arXiv preprint arXiv:2102.07113*, 2021.