Co-MLM: a SSL algorithm based on the Minimal Learning Machine

Weslley L. Caldas[1], João P. P. Gomes[1], Michelle G. Cacais[2], Diego P. P. Mesquita[1]
Computer Science Department[1]
Teleinformatics Engineering Department[2]
Federal University of Ceará, Fortaleza, Brazil
Email: {weslleylc, jpaulo, diegoparente}@lia.ufc.br, {michelle.cacais}
@great.ufc.br

***Abstract*—Semi-supervised learning is a challenging topic in machine learning that has attracted much attention in recent years. The availability of huge volumes of data and the work necessary to label all these data are two of the reasons that can explain this interest. Among the various methods for semi-supervised learning, the co-training framework has become popular due to its simple formulation and promising results. In this work, we propose Co-MLM, a semi-supervised learning algorithm based on a recently supervised method named Minimal Learning Machine (MLM), built upon co-training framework. Experiments on UCI data sets showed that Co-MLM has promising performance in compared to other co-training style algorithms.**

I. Introduction

Today, most electronic devices such as smart phones, digital cameras or smart watches provide a simple way to get multimedia data. Unfortunately, it is hard to use these data, because labeling these requires time, money and human effort. Consequently, Semi-Supervised Learning (SSL) [1], which explores the capacity of combining unlabeled data with the labeled ones, becomes an attractive methodology and has receiving attention of the community to reduce expenses.

Several approaches for SSL have been proposed in recent years, as generic models [2], graph based models [3], Semi-Supervised Support Vector Machines (S3VM) [4], and Co-training [5]. Among this approaches, Co-training is one of the most attractive ones, considered an important part of multi-vision learning, obtaining satisfactory results in different areas as computer-aided diagnosis [6], natural language processing [7] and image classification [8].

The basic idea behind the classical Co-training is to build two classifiers with two independent feature sets, which work together to select the most confident unlabeled data to predict their labels and augment the set of labeled training examples. The simplest way of creating the views is to split the original feature set into two subsets randomly. However, it is important to note that this can only be done under some assumptions and these views will be not necessarily independent [9].

The choice of the base learner is extremely important in co-training. This encouraged many co-training variations based on supervised methods that have been successful in supervised learning [10] [11] [12]. Minimal Learning Machine(MLM) is a recent method proposed by [13], that obtained promising results compared to other state-of-art methods for supervised learning, and has gained a lot of attention [14] [15] [16] for its simplicity and easiness of implementation, in addition, it has only one hyper parameter to be adjusted. In this paper, we propose a variation of co-training based on MLM, which we call Co-MLM. We validate its performance against other well-known methods based on co-training.

The rest of the article is organized as follows: in section 2, the works based on co-training are presented; minimal learning machine and the extension of MLM to SSL are presented in section 3. Section 4 presents the experimental results for both UCI and UCF-Phone data sets. In section 5, conclusions and future works are presented.

II. Review of Co-training

The standard co-training algorithm, proposed in [5], operates in two views (feature subsets) of data. In the classic co-training methodology, two learners are built using the original labeled data on each view separately. Then, each learner predicts the unlabeled samples, selects the most confident predicted examples and adds these to the training set of the other learner. After that, using the new examples provided by the other view, both classifiers are trained again. This loop will repeats until a stopping criterion or a fixed point is reached. For this approach, strong assumptions on the feature set are needed in order to guarantee the success of co-training: (i) Each view alone is sufficient to learn a good classifier; (ii) the two views are conditionally independent given the category (class label).

Given its simplicity, the co-training framework was adapted to various classifiers and achieved remarkable results. A single-view co-training method was proposed by Goldman et. al [17] named Democratic Co-learning (DCL), and does not need two independent and redundant feature sets. In DCL, an ensemble of learners is trained with original feature set of labeled data. This method use majority vote and statistical confidence interval to label the unlabeled data points and decides which examples could be added to the labeled data set. Similarly, Tritraining, was proposed by Zhou and Li et al. [18], in which three different classifiers are trained on bootstrap sampled examples.

Later, Li *et al.* proposed an algorithm called Co-Forest [6], an ensemble that uses random trees in co-training paradigm. It begins with a bootstrap sample from the original labeled data set that are used to train a set of random trees, whose will be redefined by newly selected examples at each iteration during the training process. The final prediction will be produced by majority voting.

Recently, in [19], *Liu et al.* proposed three variants of the co-training framework that could be adapted to any classifier. The variants, named multi-visions, multi-algorithms and multi-manifolds, are presented and tested in a SVM classifier and achieved promising results.

III. Minimal Learning Machine

Consider a set of N input points $X = \{\mathbf{x}_i\}_{i=1}^{N}$, with $\mathbf{x}_i \in \mathbb{R}^D$, and the set of corresponding outputs $Y = \{\mathbf{y}_i\}_{i=1}^{N}$, with $\mathbf{y}_i \in \mathbb{R}^S$. Presupposing the existence of a continuous mapping $f : \mathcal{X} \to \mathcal{Y}$ between the input and the output space, it is possible to estimate f from data with the multi-response model:

$$\mathbf{Y} = f(\mathbf{X}) + \mathbf{R}.$$

The columns of the matrices $\mathbf{X}$ and $\mathbf{Y}$ correspond to D inputs and S outputs respectively, and the rows to N observations. The columns of $N \times S$ matrix $\mathbf{R}$ correspond to the residuals.

The MLM is a two-step method designed to reconstruct the mapping existing between input and output distances and estimate the response from the configuration of the output points. In the following, the two steps are discussed.

A. Distance regression

For a selection of reference input points $R = \{\mathbf{m}_k\}_{k=1}^{K}$ with $R \subseteq X$ and corresponding outputs $T = \{\mathbf{t}_k\}_{k=1}^{K}$ with $T \subseteq Y$, define $\mathbf{D}_x \in \mathbb{R}^{N \times K}$ in such a way that its k-th column $\mathbf{d}(X, \mathbf{m}_k)$ contains the distances $d(\mathbf{x}_i, \mathbf{m}_k)$ between the N input points $\mathbf{x}_i$ and the k-th reference point $\mathbf{m}_k$. Analogously, define $\boldsymbol{\Delta}_y \in \mathbb{R}^{N \times K}$ in such a way that its k-th column $\boldsymbol{\delta}(Y, \mathbf{t}_k)$ contains the distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$ between the N output points $\mathbf{y}_i$ and the output $\mathbf{t}_k$ of the k-th reference point.

The mapping g between the input distance matrix $\mathbf{D}_x$ and the corresponding output distance matrix $\boldsymbol{\Delta}_y$ can be reconstructed using the multi-response regression model:

$$\boldsymbol{\Delta}_y = g(\mathbf{D}_x) + \mathbf{E}.$$

The columns of matrix $\mathbf{D}_x$ correspond to the K input vectors and the columns of matrix $\boldsymbol{\Delta}_y$ correspond to the K response vectors, N rows correspond to the observations. The columns of matrix $\mathbf{E} \in \mathbb{R}^{N \times K}$ correspond to K residuals.

Assuming that the mapping g between input and output distance matrices has a linear structure for each response, the regression model has the form

$$\boldsymbol{\Delta}_y = \mathbf{D}_x\mathbf{B} + \mathbf{E}. \tag{1}$$

The columns of $K \times K$ regression matrix $\mathbf{B}$ correspond to the coefficients for K responses.

The mapping matrix $\mathbf{B}$ can be estimated by minimizing the following cost function.

$$J(\mathbf{B}) = ||\mathbf{D}_x\mathbf{B} - \boldsymbol{\Delta}_y||_F \tag{2}$$

Under normal conditions, where the number of selected reference points is smaller than the number of available points (i.e., $K < N$), the matrix $\mathbf{B}$ can be approximated by the usual least squares estimate:

$$\hat{\mathbf{B}} = (\mathbf{D}_x'\mathbf{D}_x)^{-1}\mathbf{D}_x'\boldsymbol{\Delta}_y. \tag{3}$$

For an input test point $\mathbf{x} \in \mathbb{R}^D$ whose distances from K reference input points $\{\mathbf{m}_k\}_{k=1}^{K}$ are collected in the vector $\mathbf{d}(\mathbf{x}, R) = [d(\mathbf{x}, \mathbf{m}_1) \dots d(\mathbf{x}, \mathbf{m}_K)]$, the corresponding estimated distances between it's unknown output $\mathbf{y}$ and the known outputs $\{\mathbf{t}_k\}_{k=1}^{K}$ of the reference points are:

$$\hat{\boldsymbol{\delta}}(\mathbf{y}, T) = \mathbf{d}(\mathbf{x}, R)\hat{\mathbf{B}}. \tag{4}$$

The vector $\hat{\boldsymbol{\delta}}(\mathbf{y}, T) = [\hat{\delta}(\mathbf{y}, \mathbf{t}_1) \dots \hat{\delta}(\mathbf{y}, \mathbf{t}_K)]$ provides an estimate of the geometrical configuration of $\mathbf{y}$ and the reference set T, in the $\mathcal{Y}$-space.

B. Output estimation

The problem of estimating the output $\mathbf{y}$, given the outputs $\{\mathbf{t}_k\}_{k=1}^{K}$ of all the reference points and estimates $\hat{\boldsymbol{\delta}}(\mathbf{y}, T)$ of their mutual distances, can be understood as a multilateration problem [20] to estimate its location in $\mathcal{Y}$.

Numerous strategies can be used to solve a multilateration problem [21]. From a geometric point of view, locating $\mathbf{y} \in \mathbb{R}^S$ is equivalent to solving an overdetermined set of K nonlinear equations corresponding to $(S+1)$-dimensional hyper-spheres centered in $\mathbf{t}_k$ and passing through $\mathbf{y}$.

Given the set of $k = 1, \dots, K$ hypher-spheres each with radius equal to $\hat{\delta}(\mathbf{y}, \mathbf{t}_k)$

$$(\mathbf{y} - \mathbf{t}_k)'(\mathbf{y} - \mathbf{t}_k) = \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k), \tag{5}$$

the location of $\mathbf{y}$ is estimated from the minimization of the objective function

$$J(\mathbf{y}) = \sum_{k=1}^{K} \Big((\mathbf{y} - \mathbf{t}_k)'(\mathbf{y} - \mathbf{t}_k) - \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k) \Big)^2. \tag{6}$$

The cost function has a minimum equal to 0 that can be achieved if, and only if, $\mathbf{y}$ is the solution of (5). If it exists, such a solution is thus global and unique. Due to the uncertainty introduced by the estimates $\hat{\delta}(\mathbf{y}, \mathbf{t}_k)$, an optimal solution to (6) can be achieved by any minimizer $\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmin}}\, J(\mathbf{y})$ like the nonlinear least square estimates from standard gradient descent methods. In the following, the Levenberg-Marquardt (LM) method [22] is used.

C. Extension to Classification

For classification, we are still given N input points $X = \{\mathbf{x}_i\}_{i=1}^{N}$, with $\mathbf{x}_i \in \mathbb{R}^D$, and corresponding class labels $L = \{l_i\}_{i=1}^{N}$, with $l_i \in \{C_1, \dots, C_S\}$, where C_j denotes the j-the class. For $S = 2$, we have binary classification, whereas for $S > 2$ we have multi-class classification.

The MLM can be extended to classification in a straightforward manner by representing the S class labels using the 1-of-S encoding scheme. In such approach, a S-level qualitative variable is represented by a vector of S binary variables or bits, only one of which is *on* at a time. In this work, the j-th component of an output vector $\mathbf{y}_i$ is set to 1 if $l_i = C_j$ and 0 otherwise.

In the classification of a test observation $\mathbf{x}$ of unknown class label $l \in \{C_1, \ldots, C_S\}$, the estimated class $\hat{l}$ associated to the output estimate $\hat{\mathbf{y}}$ is $\hat{l} = C_{s^*}$, in which:

$$s^* = \underset{s=1,\cdots,S}{\operatorname{argmax}} \{\hat{y}^{(s)}\}, \tag{7}$$

and $\hat{y}^{(s)}$ denotes the s-th component of the vector $\hat{\mathbf{y}}$.

D. Extension to SSL: Co-Minimal Learning Machine

Recently, a generic framework for applications related to co-training was presented in [19]. It proposes three archetypes of adaptation: multi-visions, multi-algorithms and multi-manifolds, representing the great variety of existing modifications. In our approach, we use the first archetype (multi-vision) to propose a new Co-training variation based in MLM.

From a given set of N input points $X = \{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^D$, and the set of corresponding outputs $Y = \{\mathbf{y}_i\}_{i=1}^N$, with $\mathbf{y}_i \in \mathbb{R}^S$. Suppose the data set $\hat{X} = L \cup U$, which $L = \{(x_i, y_i)\}_{i=1}^l \in X \times Y$ are the labeled data and $U = \{(x_i)\}_{i=l+1}^n \in X$ are the unlabeled data. The l first examples of $\hat{X}$ are the labeled data and the $u = n - l$ last examples are unlabeled data, both contained in the training set.

First, the data set $\hat{X} \in \mathbb{R}^{N \times D}$ is randomly split into two mutually exclusive subsets of features V_1 and V_2 with equal sizes, representing two views in which $V_j \in \mathbb{R}^{N \times \frac{D}{2}}$ So, two weak classifiers h_1 and h_2 with different visions are built utilizing the original labeled data set. For each classifier, a k set of reference points are chosen using *k-medoids* algorithm [23]. Since the subset of features V_1 and V_2 are different, it is expected that different *medoids* would be chose, thereby increasing the diversity of classifiers.

Then, each classifier labels all unlabeled data, and together they select the most confident example for each class to insert into the labeled data set. To choose the best examples, we have utilized weighted output of both classifiers based on the previous classifications schema. In this case, $s*$ is calculated as follows:

$$s^* = \underset{s=1\ldots S}{\operatorname{argmax}} \sum_{j=1}^{2} [W_j * h_{j,s}(V_j(x))] \tag{8}$$

In which the vector $\hat{y}_{(j)} = [\hat{y}_{(j)}^{(1)}, \hat{y}_{(j)}^{(2)}, ..., \hat{y}_{(j)}^{(S)}]$ denotes the output estimate of the j-th classifier in a 1-of-S output encoding, $V_j(x)$ is the vision j for a observation x, $h_{j,s}(V_j(x))$ is the output provided by the classifier h_j for the class s for this observation, that is $\hat{y}_{(j)}^s$, and W_j is a weight factor provided on the accuracy of the original labeled data for h_j. The observations with the biggest $s*$ for each class are selected to put in new training data set. This measure was adopted to put more weight in the decision of the classifier that had obtained better results, preventing noise accumulation in the training phase.

After this process, the two weaker classifiers are trained again, using the augmented training set, with the labeled data of the previous interaction. This process repeats until both classifiers converge or a number of retries is reached.

Finally, we will build a strong classifier using the full feature set and all labeled data provided by $H = \{h_1, h_2\}$, which was acquired in co-training phase after t-th iterations.

Algorithm 1 *Co-MLM*

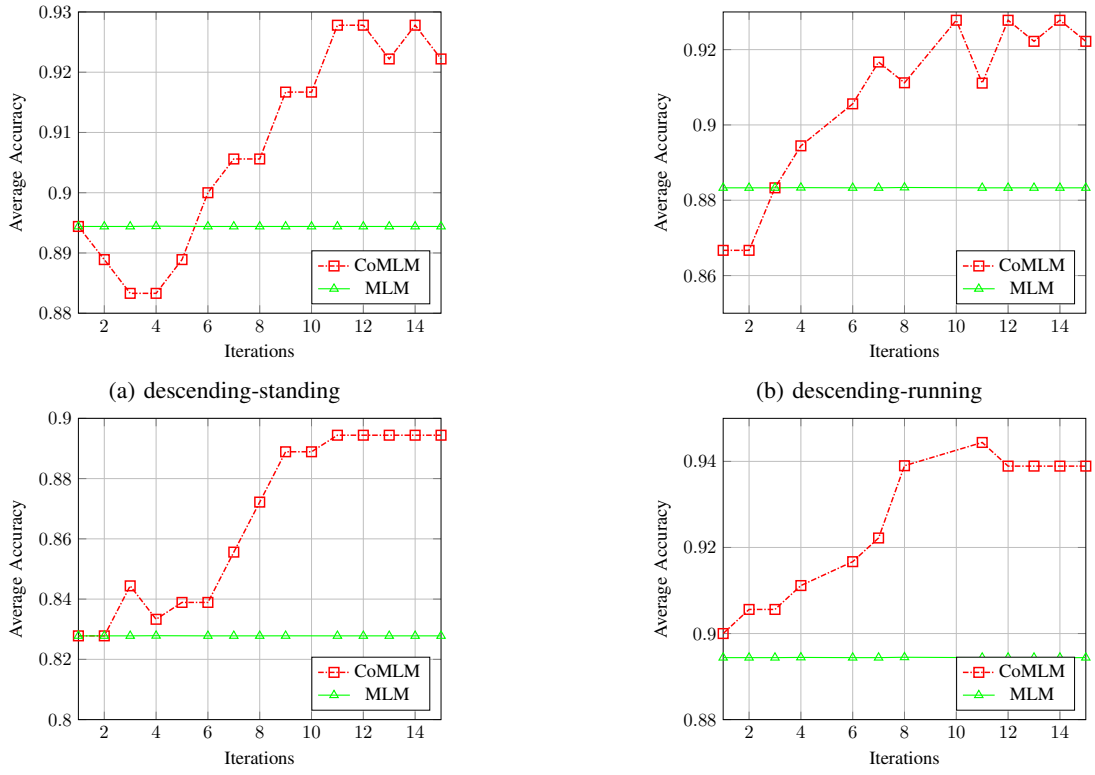
procedure FIND $H = \{h_1, h_2\}$
Inputs: training set $\hat{X} = L \cup U$, where *L* contains *l* labeled training examples and *U* contains *u* unlabeled training examples, and number of k reference points.
Outputs: $H = \{h_1, h_2\}$, *l* labeled examples.
  repeat
    for j=1...2 **do**
      Build V_j from L^i;
      Use *K-medoids* to select K reference points, R, from V_j and their corresponding outputs, T, from Y, only from labeled data;
      Compute D_x: The distance matrix between V_j and *R*;
      Compute Δ_y: The distance matrix between *Y* and *T*;
      Calculate $\hat{B}_j = (D_x^{'} D_x)^{-1} D_x^{'} \Delta_y$.
      Compute $\hat{\delta}(\mathbf{y}, T) = d(x, R)\hat{B}_j$;
      Use T e $\hat{\delta}(\mathbf{y}, T)$ to find an estimate for $\mathbf{y}$, for all unlabeled data in U.
      $h_j = \{B_j, R\}$
    end for
    Calculate W_j for $h_j (j = 1, 2)$.
    Select 1 example for each class, using the rule (8), and form the new training set, L^{i+1}.
  until $\{$*repetitions*$\}$ or $\{$*convergence*$\}$ or $\{U = \emptyset\}$
Return: $H = \{h_1, h_2\}$
end procedure

IV. Experiments and Results

This section is composed for descriptions and resolutions of two experiments. In the first one, we utilized UCF-DataPhone, a real data set with two sufficient and different visions, to illustrate the capacity of Semi-Supervised Learning of our methodology in a real environment. After, we used 8 data sets from the UCI repository, in order to determine the general performance of Co-MLM compared to other methods of the state of art.

A. UCF-iPhone dataset

UCF data set is a set of aerobic actions (biking, climbing, descending, exercise biking, jump roping, running, standing, walking and treadmill walking) provided by the University of



(a) descending-standing

(b) descending-running

(c) climbing-running

(d) climbing-jumping

Fig. 1: Comparison between MLM and Co-MLM about the number of interactions in Co-training phase. Initially, both MLB and CoMLM are trained with 60% of labeled data (54 examples, 27 per class), in each CoMLM interaction is refined adding 2 examples (one per class) and retrained.

Central Florida, recorded from subjects using Apple iPhone 4 smart phone [24]. It was utilized the Inertial Measurement Unit (IMU) with a 3D accelerometer (accelerometer), angular velocity (gyroscope), and orientation (magnetometer) for obtaining data of movements. The samples was taken at 60Hz, and manually trimmed to 500 samples, then transformed into instance with 1500 features representing one action.

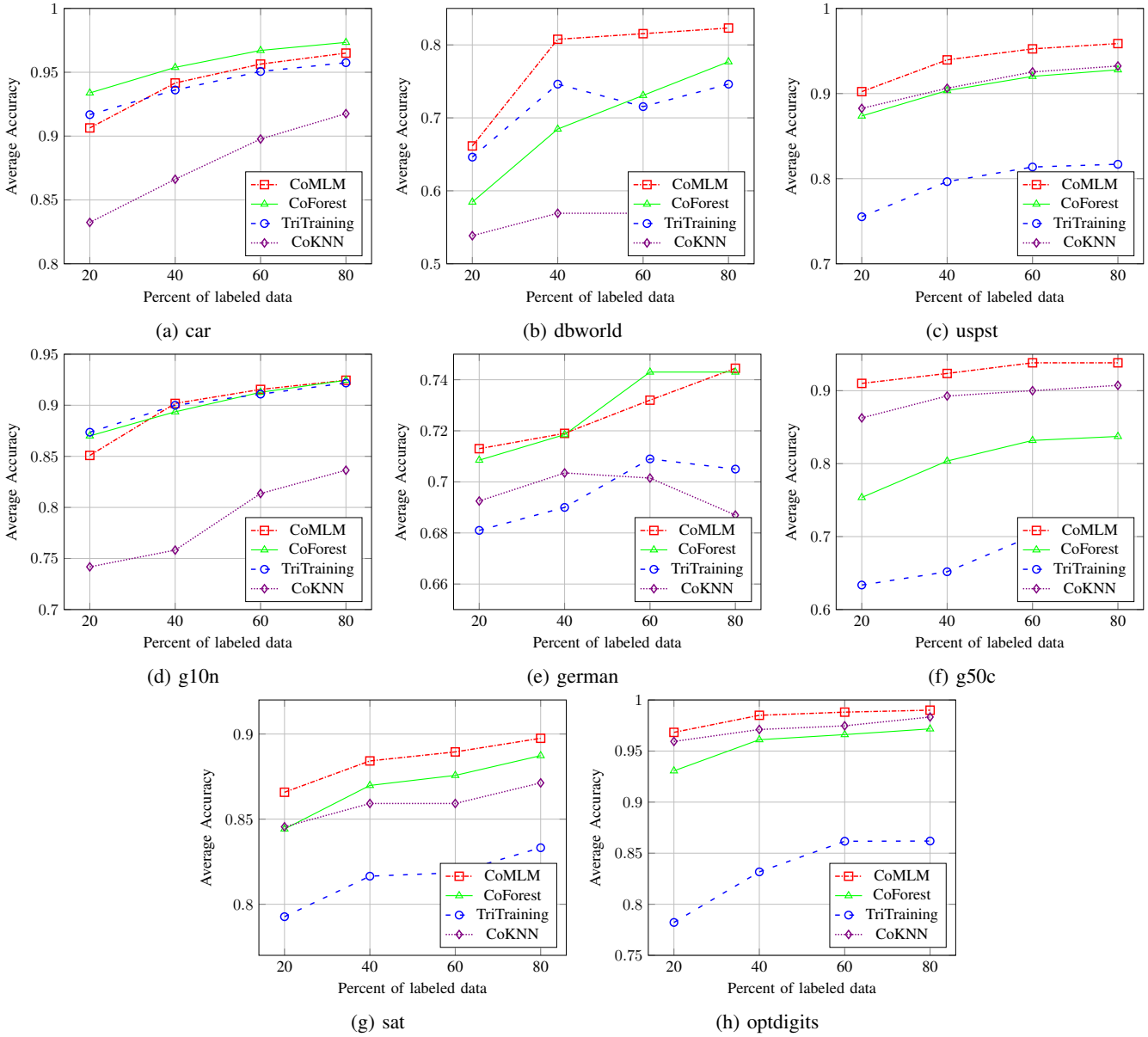
In our experiments, we selected the 6 better represented classes of accelerometer and gyroscope data (45 examples per class), and done a procedure one-vs-one with this classes, amounting 15 different data sets. The data collected by the accelerometer and gyroscope could be understood as two distinct and sufficient views. Then, 4 data sets with great accuracy (conditions of sufficiency) and a small mutual information conditioned to the class (condition of independence) between the visions were selected to be used in our experiments. We had calculated the mutual information according to [10]. it was adopted 10 k-folds cross validation strategy for each data set. Each fold was divided into three sets: train, test and validation, and each set contained at least one instance for class. We used 60% of the labeled data of training set to form L and the remainder to form U. Our intention is to investigate Co-MLM learning in a real environment as interactions advance.

Figures 1a, 1b, 1c and 1d, demonstrate comparison between Co-MLM and MLM. It is easy to see that in a general way, the error gradually decreases to the extent that interactions advance, showing that under right conditions, Co-MLM presents a considerable better performance compared to MLM. It is expecting accuracy variation during interactions, since wrong data could be added in a interaction, however it is important to note that as the interactions advance, the data correctly classified tend to offset the noise accumulation.

B. UCI Data sets

Initially, 8 data sets from UCI repository [25] with different sizes and number of features were chosen.It was adopted the same methodology in the last experiment, however to form the set $\hat{X} = L \cup U$, it was used 20%, 40%, 60% and 80% of training data to compose L, and the rest to compose U. Data formatting can be seen in the Table I.

It have been chosen 3 semi-supervised learning well known methods based on Co-training: Co-forest [6], Tritraining [18] and standard Co-training [5]. The base classifiers of Tritraining and co-training were respectively j48 and knn. To promote a fair comparison, all the hyper parameters have been adjusted according to validation set: to Co-forest, the number of classifiers and the confidence interval; to Tritraining (j48), the confidence interval and the number of instances per leaf; and to Co-training (Knn), the number of nearest neighbors for both classifiers. Co-forest and Tritraining were provided utilizing



(a) car (b) dbworld (c) uspst

(d) g10n (e) german (f) g50c

(g) sat (h) optdigits

Fig. 2: The average precision of different methods and percent of labeled data.

TABLE I: Description of data sets.

Data set	Instances	Features	Class
car	1724	6	4
g10n	548	10	2
german	998	24	2
sat	1994	36	6
dbworld	62	4702	2
g50c	548	50	2
optdigits	1787	64	10
uspst	1997	256	10

WEKA data mining tool [26]. Co-MLM and Co-training were implemented in MATLAB, since the purpose of this article is to ascertain accuracy, not speed, the difference in language does not interfere the evaluation.

The number of repetitions and instances selected by interaction are important factors for good performance for methods based on co-training. A few repetitions could add no relevant information, while a lot of repetitions could add noise. In our experiment, we choose utilizing 5 interactions for Co-MLM. A small number of interaction was chosen to avoid adding noise to data. The number of the selected examples was one example per class. Co-forest and Tritrainig does not use a preestablished number of interactions, their process only finish when some internal classifiers presents any significant change, while the number of added samples due to a threshold. In this experiment, Co-training also utilize 5 interactions, selecting 1 example of each class for each classifier, forming a total of up to two examples per class for each interaction.

As expected, Tritraining and Co-training obtained less satisfactory results in relation to Co-MLM and Co-Forest, since those are simpler methods. It is possible to observe bad results for Co-training when the data sets does not contain a lot of features, as seen in Figures 2a, 2d and 2e or a sufficient initial accuracy, Figure 2b. This paradigm change in Figures 2c, 2f, 2g and 2h. It is expected more independence among the visions with many features, and with high initial accuracy, the classifiers could be enough to produce good results. The same factor occurs to Co-MLM, but it demonstrated been more robust, once it got better results than Co-training in data sets with a few features, as seen in Figures 2a, 2d and 2e, or accuracy 2b. Co-forest presented, in general, similar results than Co-MLM, but once Co-forest is an ensemble method, many classifiers are requested, while for Co-MLM, only two are required, ordering less computational effort. Tritraining presented certain irregularities, as seen in Figures 2e, 2f and 2b, in which increasing the number of labeled data detracted the performance. It could be explained with the number of labeled data: the bigger it is, the bigger will be the trust rating and more examples will be added; but not necessarily there are correct examples. This shows the great irregularity in SSL [3] [27]. It is worth noting however, that CoForest and Co-MLM not presented this irregularities.

V. Conclusion

This paper proposed an adaptation of Minimal Learning Machine for the semi-supervised paradigm. Empirical analysis validates Co-MLM's learning capabilities for unlabeled data and performance gain of Co-MLM regarding MLM under certain conditions.

Furthermore, We showed that Co-MLM presents an attractive alternative to other state-of-the-art SSL algorithms, presenting similar performance and having only one hyper parameter to be tuned.

However, Co-MLM may present noise accumulation in co-training phase. Best ways to select the most confident observations will be proposed in future works.

Acknowledgments

The authors acknowledge the support of CNPq (Grant 402000/2013-7 and Grant 456837/2014-0).

References

[1] X. Zhu, "Semi-supervised learning literature survey," 2005.

[2] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[3] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," 2001.

[4] K. Bennett, A. Demiriz *et al.*, "Semi-supervised support vector machines," *Advances in Neural Information processing systems*, pp. 368–374, 1999.

[5] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[6] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 6, pp. 1088–1098, 2007.

[7] D. Pierce and C. Cardie, "Limitations of co-training for natural language learning from large datasets," in *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001, pp. 1–9.

[8] J. Yu, M. Wang, and D. Tao, "Semisupervised multiview distance metric learning for cartoon synthesis," *Image Processing, IEEE Transactions on*, vol. 21, no. 11, pp. 4636–4648, 2012.

[9] U. Brefeld, "Multi-view learning with dependent views," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 2015, pp. 865–870.

[10] F. Feger and I. Koprinska, "Co-training using rbf nets and different feature splits," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE, 2006, pp. 1878–1885.

[11] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 235–243.

[12] K. Li, J. Zhang, H. Xu, S. Luo, and H. Li, "A semi-supervised extreme learning machine method based on co-training," *J. Comput. Inf. Syst*, vol. 9, no. 1, pp. 207–214, 2013.

[13] S. J. A.H., C. F., M. Y., L. A., B. G., and S. O., "Minimal learning machine: A new distance-based method for supervised learning," in *12th International Work Conference on Artificial Neural Networks (IWANN'2013)*, 2013, pp. 408–416.

[14] D. P. P. Mesquita, J. P. P. Gomes, and A. H. S. Jr., "A minimal learning machine for datasets with missing values," in *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I*, 2015, pp. 565–572.

[15] D. P. P. Mesquita, J. P. P. Gomes, and A. H. S. Junior, "Ensemble of minimal learning machines for pattern classification," in *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part II*, 2015, pp. 142–152.

[16] A. S. C. Alencar, W. L. Caldas, J. P. P. Gomes, A. H. d. Souza, P. A. C. Aguilar, C. Rodrigues, W. Franco, M. F. d. Castro, and R. M. C. Andrade, "Mlm-rank: A ranking algorithm based on the minimal learning machine," in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, Nov 2015, pp. 305–309.

[17] Y. Zhou and S. Goldman, "Democratic co-learning," in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. IEEE, 2004, pp. 594–602.

[18] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 11, pp. 1529–1541, 2005.

[19] W. Liu, Y. Li, D. Tao, and Y. Wang, "A general framework for co-training and its applications," *Neurocomputing*, vol. 167, pp. 112–121, 2015.

[20] M. M. Niewiadomska-Szynkiewicz, Ewa, "Optimization schemes for wireless sensor network localization," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 291–302, 2009. [Online]. Available: http://eudml.org/doc/207936

[21] W. Hereman, Ü. Göktaş, M. D. Colagrosso, and A. J. Miller, "Algorithmic integrability tests for nonlinear differential and lattice equations," *Computer Physics Communications*, vol. 115, no. 23, pp. 428 – 446, 1998, computer Algebra in Physics Research.

[22] 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. [Online]. Available: http://dx.doi.org/10.1137/0111030

[23] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[24] C. McCall, K. K. Reddy, and M. Shah, "Macro-class selection for hierarchical k-nn classification of inertial sensor data."

[25] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[27] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.