CoMet: A Meta Learning-Based Approach for Cross-Dataset Labeling Using Co-Training

Extended Abstract

Zaks Guy
Ben-Gurion University
Beer-Sheva, Israel
zaksg@post.bgu.ac.il

Katz Gilad
Ben-Gurion University
Beer-Sheva, Israel
giladkz@bgu.ac.il

ABSTRACT

In many practical domains, applying machine learning is challenging not due to the lack of available data, but because labeled samples are in short supply. A common approach for obtaining additional labeled samples is *co-training*, a semi-supervised learning setting where two learners (agents), trained on different perspectives of the data, iteratively label additional samples. The rationale of this approach is that the different learner perspectives will produce a more diverse labeled set, resulting in more effective classifiers. While co-training proved effective in multiple cases, the labeling mechanisms used by existing approaches are heuristic and error-prone. We propose CoMet, a meta learning-based co-training algorithm. CoMet utilizes meta-models trained on previously-analyzed datasets to select the samples to be labeled for the current dataset. Our experiments, conducted on 35 datasets, show that CoMet significantly outperforms the standard co-training approach.

KEYWORDS

Machine learning; Multi-agent learning; Co-training; Meta-learning; Semi-supervised learning; Data-labeling; Cross-dataset

ACM Reference Format:
Zaks Guy and Katz Gilad. 2020. CoMet: A Meta Learning-Based Approach for Cross-Dataset Labeling Using Co-Training. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Labeled data is often difficult to obtain. This difficulty can stem from the labeling process being expensive, long, or requiring the involvement of a human expert. In such cases, it is common to leverage *semi-supervised* machine learning solutions. Semi-supervised learning can extract meaningful insights from unlabeled data and then leverage these insights to supervised learning problems.

A prominent semi-supervised learning approach designed to increase the number of labeled samples is *co-training*, originally proposed by Blum and Mitchell [1]. This approach consists of training two agents (i.e., learning algorithms) on different views of the available labeled data. Each agent then selects a small set of unlabeled samples of whose classification it has a high degree of certainty, and adds them to the labeled set. The intuition behind this approach is that by creating two different "perspectives" of the data, each agent will select samples the other agent would not, thus increasing the diversity of the labeled training set and preventing over-fitting. The basic assumption for applying co-training methods is the *sufficient and redundant views* assumption [1], however recent studies [4, 10] demonstrated that co-training can be effective even when the conditions are not met.

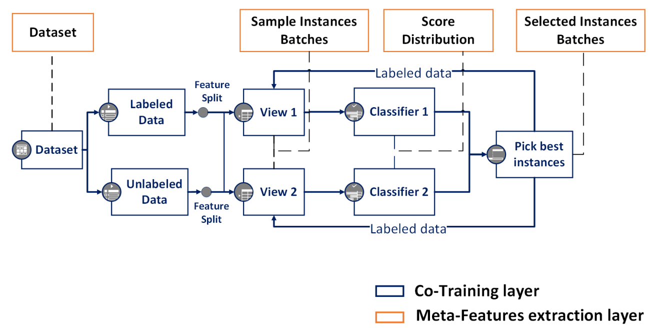

Figure 1: The meta-features extraction points during the co-training process. The co-training layer represents the original co-training algorithm. The meta-features extraction layer collects data regarding the dataset, the classifiers, the batches and their instances.

All existing co-training algorithms generally apply the same heuristic approach for sample labeling: each agent selects a small number of samples in which it has the highest confidence and adds them to the labeled set. This approach has several drawbacks: *(a)* the samples are selected individually, without any consideration of the characteristics of the overall *batch of samples*; *(b)* there is no attempt to determine whether the performance of the two agents is correlated, a fact that should theoretically impact their labeling strategy, and; *(c)* the characteristics of the analyzed dataset (e.g., size, values distribution) are in no way taken into account.

Co-training style methods can be applied on NLP tasks [2, 5], image classification [6, 7] and tabular data classification [9, 10]. We trained and tested CoMet on tabular data for binary classification, based on 35 different datasets from the OpenML repository (www.openml.org). Our contributions are as follows:

- We propose a co-training algorithm that applies meta learning instead of simplistic heuristics. Additionally, we propose a batch-selection approach that is better than the single-sample selection used today.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

- Extensive evaluation demonstrates that our approach significantly outperforms the standard co-training algorithm.

2 THE PROPOSED METHOD

In this study, we propose CoMet, a meta learning-based approach for co-training. Our approach models the analyzed dataset (both labeled and unlabeled datasets) and the performance of each agent and trains a learning model select the samples that will be added to the labeled set of samples. In addition to being able to dynamically adapt its sample-selection policy to the analyzed dataset, CoMet also aims to optimize its *batch selection*, i.e. make sure that the set of selected samples complement each other and the performance of the co-training algorithm. Additionally, this study is the first (to the best of our knowledge) to leverage information from previously-analyzed datasets in order to select the added samples in the co-training process. Our proposed framework is presented in Figure 1. Our overarching goal is to create a co-training algorithm that *(a)* aims to select useful batches of samples rather than unrelated individual samples, and; *(b)* leverages meta-learning methods to make the batch-selection process dynamic and adaptable. To this end we employ three stages:

1. Meta-model creation (the "offline" phase): We run multiple co-training experiments on a large set of datasets in order to ascertain what makes a batch effective. The chosen datasets have large variance in their characteristics–number of samples, number of features, feature type composition, etc.—designed to make the resulting meta-model as robust and generic as possible. For each batch/dataset combination, we extract multiple meta-features and paired them with the batch's contribution to the performance on the dataset's test set. These meta-features are then used to train our meta-model. The goal of this phase is to generate a set of meta-features that will enable our meta-model to rank the candidate batches based on their effectiveness. To achieve this goal, we generate four types of meta-features: (a) *dataset-based meta-features*; (b) *confidence-score distribution meta-features*; (c) *batch-based meta-features*; and (d) *instances-based meta-features.* The extracted meta-features describe the dataset characteristic, the classifiers' confidence score distribution, features correlation and diversity, comparison to previous iterations and provide a "sneak peek" into the future to gauge the effect that the candidate batch will have on the classifiers' behaviour, using a temporal-difference (TD) [3, 8] style meta-features.

2. Candidate batches generation: For every iteration of the co-training process, we generate a large and diverse set of candidate batches, from a fixed percentage of the top-ranked samples. We generate $\binom{4}{2}^4 = 1296$ batches for each co-training iteration. We reach this given that our datasets have two classes (i.e., binary datasets), four subsets of samples (i.e., class-partition combinations), and we chose to perform this process four times.

3. Candidate batch ranking and selection (the "online" phase): Out of all generated batches, we need to select the batch with the highest improvement potential (i.e., increase the AUC score of the test set). For each candidate batch, we extract the meta-features that enable our pre-trained meta-models (stage 1) rank all candidates and select the best batch. Then we add its samples to the labeled set. The updated labeled set will form the ground truth for the next co-training iteration.

3 EXPERIMENTS

The goal of our evaluation is to determine whether our batch and meta learning-based approach outperforms the standard co-training algorithm. Our evaluation includes the overall performance – the final classification performance of the respective algorithms. We used the following setting throughout the evaluation:

- The labeled set size was set to 100 samples, all sampled randomly from the training set. The remainder of the training set was used as the unlabeled set.
- We used the random forest algorithm and sorted its classification confidence score to rank the batches. The algorithm used by our two classifiers H_1, H_2 was logistic regression.
- Both CoMet and the standard co-training algorithm selected eight samples at each iteration – four samples by each classifier (in accordance with [1]).
- For each instance of the dataset used to train the meta-model, a batch was labeled as 'positive' if it improved the AUC metric by a value of at least 0.005 and 'negative' otherwise.
- We used a leave-one-out (LOO) approach for the training of the meta-model: for each evaluated dataset d_i, we trained the meta-model using meta-features from all batches.
- For every dataset we ran 10 co-training experiments of 20 iterations each. The initial seed of labeled samples was randomly chosen for each experiment, and both CoMet and the standard co-training algorithm used the same seeds.
- We used the error reduction as the evaluation metric.

For each dataset, we compare the error rate achieved by the standard co-training algorithm to that achieved by our approach. CoMet significantly outperforms the baseline algorithm, achieving an average error reduction rate of 5.2%. Table 1 presents the error reduction comparison between CoMet and the original co-training method, split based on the datasets for which CoMet did and did not show an improvement in performance. The significance of our results was verified using a paired sample T-test, with $p \leq 0.01$. Additionally, our analysis indicates that the smaller the initial AUC score (i.e., before any co-training learning), the better CoMet's relative performance compared to the original co-training method.

Table 1: The relative performance of CoMet compared to the original co-training algorithm. We show overall statistics and a split based on whether or not CoMet showed improvement.

	Number of datasets	CoMet error reduction rate
Improved performance	29 (83%)	6.7%
Reduced performance	6 (17%)	-1.8%
Total	**35**	**5.2%**

4 CONCLUSIONS

In this paper we represent CoMet, a meta learning-based co-training algorithm. Our approach focuses on selecting batches of useful samples rather than individual samples, and uses meta-learning to model the possible impact of the analyzed batches on the training data and the learning models that take place in the co-training process.

REFERENCES

[1] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*. Citeseer, 92–100.

[2] Jason Chan, Irena Koprinska, and Josiah Poon. 2004. Co-Training on Textual Documents with a Single Natural Feature Set.. In *ADCS*. Citeseer, 47–54.

[3] Pawel Cichosz. 1994. Truncating temporal differences: On the efficient implementation of TD (lambda) for reinforcement learning. *Journal of Artificial Intelligence Research* 2 (1994), 287–318.

[4] Gilad Katz, Cornelia Caragea, and Asaf Shabtai. 2018. Vertical Ensemble Co-Training for Text Classification. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 2 (2018), 21.

[5] Donghwa Kim, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. 2019. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences* 477 (2019), 15–29.

[6] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. 2018. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*. 135–152.

[7] Sathishkumar Samiappan and Robert J Moorhead. 2015. Semi-supervised co-training and active learning framework for hyperspectral image classification. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 401–404.

[8] Gerald Tesauro. 1995. Temporal difference learning and TD-Gammon. *Commun. ACM* 38, 3 (1995), 58–68.

[9] Yan Zhou and Sally Goldman. 2004. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 594–602.

[10] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge & Data Engineering* 11 (2005), 1529–1541.