Robust multi-label learning for weakly labeled data

Atanas Marinov[1] , **Amirmasoud Ghiassi**[1] , **Taraneh Younesian**[1] , **Lydia Chen**[1]

[1]TU Delft

Abstract

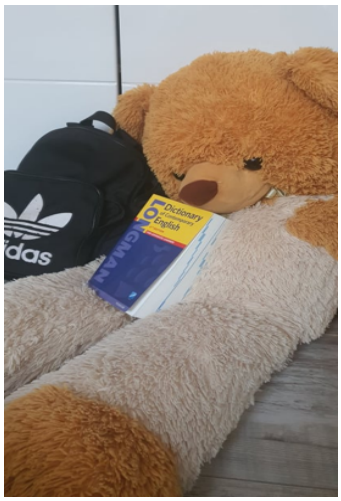
Multi-label learning is one of the hot problems in the field of machine learning. The deep neural networks used to solve it could be quite complex and have a huge capacity. This enormous capacity, however, could also be a negative, as they tend to eventually overfit the undesirable features of the data. One such feature presented in the real-world datasets is imperfect labels. A particularly common type of label imperfection is called weak labels. This corruption is characterized not only by the presence of all relevant labels but also by the addition of some irrelevant ones. In this paper, a novel method, Co-ASL, is introduced to deal with the label noise in multi-label datasets. It combines the state-of-the-art approach for multi-label learning, ASL, with the famous robust training strategy, Co-teaching. The performance of the method is then evaluated on noisy versions of MS-COCO to show the lack of overfitting and the performance improvement over the non-robust multi-label ASL.

1 Introduction

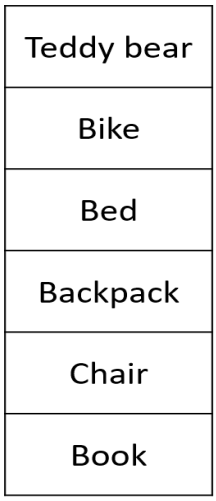
Currently, most of the machine learning work done is in the field of supervised learning. Its goal is to predict a function that maps an input to output (label) from a predefined set of output labels based on already provided examples of input-output pairs. Multi-label learning (MLL) is an extension of supervised learning, in which for each data input, a subset of the predefined set of output labels is assigned. Therefore, MLL aims to be able to assign more than one label per input. This small change in the traditional concept turns out to be very powerful and could potentially improve quite dramatically the performance of users of the resulting output labels. That is mainly because the number of dimensions of the data we use constantly grows and the fact that most of the nowadays subjects are complicated, and their meaning could depend on more than of its aspects.

However, in the real world, it also often happens that example input-output pairs, provided to supervised learning, and in our case to MLL have some imperfections. For example, both crowdsourcing [18] and online queries[11] are famous for the noise they generate. Reportedly, real-world datasets contain noise ranging between 8% and 38% [9]. There are 3 core types of label noise: i) weak labels - all relevant but also some irrelevant labels are present, ii) wrong labels - some of the relevant labels are replaced by irrelevant ones, and iii) missing labels - some of the relevant labels are not presented. The method proposed will be designed to deal specifically with weak labels (Figure 1).

The main problem which is tried to be solved is that modern deep neural networks, that are used to do MLL, have a big capacity, and they can easily overfit those corruptions. Therefore,



(a) Example input image



(b) Example label set

(c) True labels

(d) Weak labels

Figure 1: An illustrative example of MLL and weak labels. (a) is an example input image that could be part of a multi-label dataset. (b) is the set of possible output labels - it contains all the labels that could potentially be assigned to the pictures in the dataset. (c) is the set of true (relevant) labels for the image example in (a) - the image only contains 3 out of the 6 labels in the label set - teddy bear, backpack, and book. The other 3 are not presented and are, therefore, excluded from (c). (d) is a possible weak label configuration – the image is assigned to 5 out of 6 possible labels. The weak label set includes all 3 true labels but also 2 that are not presented on the image - bed and bike.

the result will be a poor generalization error, and more importantly, a bad real-world performance.

There is quite a lot of research in both fields - MLL and learning from noisy examples. The novel method presented in this paper (Co-ASL) solves the problem by combining the most prominent solutions in the two fields - ASL[2], the state-of-the-art method used for multi-label classification, and Co-teaching[7], an eminent approach to deal with label noise in the single-label classification. Similar to Co-teaching, Co-ASL achieves robustness by using two networks that evaluate small-loss instances for every mini-batch and then exchange those in between to update their weights.

In the next section, some of the work in the two relevant fields will be discussed. Following it, a more in-depth description of Co-teaching, ASL, and Co-ASL will be given. Subsequently, an evaluation of the performance of the proposed novel method will be conveyed. During it, the algorithm will be tested against the plain ASL on different amounts of noise. The paper will then finish with a review of the ethical side of the research.

2 Related Work

Multi-label learning

MLL is a hot topic in the field of machine learning and there is already a variety of approaches tackling the MLL problem. Some exploit the label correlation using graph neural networks [4], others modeled the attentional regions of the image [20]. ASL solves the problem without adding any additional complications to the architecture of the network and instead by modifying the loss function.

Single-label learning from noisy labels

Similarly, quite a bit of time has been put into robust algorithms that could deal with noisy labels and specifically in filtering them. The approaches to solve those can be subdivided into 3 categories [17] - Multi-network Learning, Multi-round Learning, and Hybrid approach. The multi-network approach is realized using collaborative learning and co-training. Concrete implementations of it are

Co-teaching, Co-teaching+ [21], MentorNet [8] and Decoupling[12]. MentorNet uses two networks. It pre-trains the first one and uses it as a mentor to supervise the training of the other one. Decoupling also uses two networks, but they are trained simultaneously and update their weights only with the instances that have different predictions from the two networks. Similarly to Decoupling, Co-teaching+ uses "Update by Disagreement" but on top of Co-teaching.

In contrast, Multi-round learning does not need an extra DNNs, but it iteratively refines the selected set of examples. An example of multi-round learning is ITLM [16]. The hybrid approach tries to deal with the problem that the sample selection discards all of the non-selected examples from the training data. One of the most predominant instances of such a method is SELF [13]. The issue with all those methods is that they are tailored for only one of the two aspects of the problem - MLL or label corruption, and none of them are designed to deal with noisy labels in the multi-label case.

Multi-label Learning from noisy labels

In contrast to the previous, not enough attention has been put in the combined field. [24] critiques the exploitation of label correlations in the multi-label classifiers as they lead to poor generalization error and introduces Context-Based Multi-Label Classifier (CbMLC) - a framework that leverages word embeddings to perform regularization for multi-label classification. [19] proposes an approach that trains multi-label classifier and noisy label identifier simultaneously to recovers the ground-truth labeling information for partial labels. [23] also tackles the partial label problem but solves it by calculating the confidence of the candidate label and then utilizing the credible labels with high labeling confidence.

3 Methodology: Co-ASL

This section contains a description of the method proposed (Co-ASL) to solve the Multi-label learning problem with weak labels. First Co-teaching will be introduced more formally, followed by ASL. The chapter will then end with a description on how to combine the two.

3.1 Co-teaching

Co-teaching will be used to deal with the label noise. It is one of the distinguished methods for that and achieves great performance for the single-label problem. It doesn't put constraints on the networks that it uses, which allows it to be easily integrated into different methods. Similarly to Co-training [3], Co-teaching uses two deep neural networks, which are trained simultaneously. The training begins with the initialization of the two models, f and g, with their input hyperparameters, w_f and w_g. Then the training set $\mathcal{D}$ gets shuffled, and for each epoch T, it gets fed in a mini-batch manner. For each minibatch $\overline{\mathcal{D}}$ calculate the loss of every example. Based on those losses form the small-loss instances $\overline{\mathcal{D}}_f$ and $\overline{\mathcal{D}}_g$, which contain only $R(T)$ percentage of the inputs in $\overline{\mathcal{D}}$ with the smallest loss for f and g, respectively. Then f and g update their weights using only the small-loss instance of their peer, i.e., w_f gets updated using $\overline{\mathcal{D}}_g$ and w_g get updated using $\overline{\mathcal{D}}_f$. After the update of the weights finishes, the small-loss percentage $R(T)$ also gets adjusted. The process is then repeated for every epoch.

Co-teaching works because of the "memorization" effect[1][22] that deep neural networks have, i.e., the easy examples are getting learned first. It often happens that the non-corrupted examples in a noisy dataset are easier to learn compared to the wrong ones. Because of that, the small-loss instance is more likely to contain mainly correct examples. That way, the irrelevant instances are getting "filtered out" from being used during the early training stages. However, in the later ones, when the number of epochs gets large, and the networks eventually overfit those corruptions. To solve that, a dropout rate $R(T)$ is introduced - in the early epochs, keep the small-loss instances bigger and therefore allow the networks to learn the correct labels. Then as the epochs progress, increase $R(T)$ and by that the filtering respectively, i.e., decrease the size of the small-loss instances, to prevent the networks from overfitting to label noise.

3.2 ASL

ASL is chosen to tackle the MLL part of the problem. It achieves a state-of-the-art

MLL performance without complicating the architecture of the deep neural network. That way, it allows being easily extended with Co-teaching. It uses Binary Cross-Entropy loss extended with Asymmetric Focusing and Asymmetric Probability Shifting. Asymmetric Focusing decouples the focusing levels of positive and negative examples by introducing separate parameters for them, γ_+ and γ_+, respectively. Asymmetric Probability Shifting reduces the impact of the negative examples with very low probabilities to the loss by performing hard thresholding, i.e., it fully discards them. As a result, the shifted probability p_m is defined as:

$$p_m = max(p - m, 0)$$

, where m denotes the probability margin and is a tunable hyper-parameter. After merging the Binary Cross-Entropy loss with Asymmetric Focusing and Asymmetric Probability Shifting Asymmetric Loss Function (ASL) is defined as:

$$l_{ASL} = \begin{cases} L_+ = (1-p)^{\gamma_+} \log(p) \\ L_- = (p_m)^{\gamma_-} \log(1-p_m) \end{cases}$$

3.3 Combining Co-teaching and ASL

As both Co-teaching and ASL do not put any additional requirements and complications on the deep neural networks used, the conversion of ASL to Co-ASL is straightforward. Pseudocode illustrating the implementation of Co-ASL is shown in Algorithm 1. Co-ASL follows all of the steps of Co-teaching, described in section 3.1. The only change that needs to be done is to replace the loss function that Co-teaching is using with ASL (lines 5,6,8,9 in Algorithm 1).

Algorithm 1: Co-ASL

1 **Input:** w_f and w_g, learning rate lr, number of epoch T_{max}, $\mathcal{D}$
2 **Shuffle** training set $\mathcal{D}$
3 **for** T = *1,2,...,*T_{max} **do**
4 **for** $\overline{\mathcal{D}}$ *in* $\mathcal{D}$ **do**
5 **Calculate** $l_{ASL}(f, \overline{\mathcal{D}})$
6 **Calculate** $l_{ASL}(g, \overline{\mathcal{D}})$
7 **Obtain** $\overline{\mathcal{D}}_f$ and $\overline{\mathcal{D}}_g$
8 **Update** $w_f = w_f - lr \nabla l_{ASL}(f, \overline{\mathcal{D}}_g)$
9 **Update** $w_g = w_g - lr \nabla l_{ASL}(f, \overline{\mathcal{D}}_f)$
10 **end**
11 **Update** $R(T)$
12 **end**

4 Evaluation

4.1 Setup

Dataset

The dataset used for the experiments on Co-ASL is MS-COCO[10]. It is one of the most popular choices for multi-label image classification and object detection. To focus more specifically on the evaluation of the multi-label case, all the instances with less than 2 labels were removed from the training and the validation set. Due to time and hardware constrains, only a third of the training set, or around 21K examples, was used for training.

Label noise

MS-COCO is a clean dataset, i.e., neither the label in the training set nor the ones in the validation one have imperfections. To evaluate the performance of Co-ASL, an artificial noise is injected into the training set. The type of noise injected is symmetric, similar to the one in some of the related work[14]. The amount of noise injected is determined by the corruption rate - for each label presented, there is a uniform probability to turn one of the other non-presented labels into presented. For example, if an instance with 4 true labels is taken and the noise injected has a 50% corruption rate, the expected number of corrupted labels is 2, resulting in a total of 6 labels expected after the noise injection. However, because of the randomness involved, this total expected number is not fixed.

Baseline

The baseline used for comparison is the pure ASL version provided by its creators. Both ASL and Co-ASL are using TResNet[15] as an underlying neural network architecture. TResNet is a high performance architecture based on ResNet. The one used is pre-trained on the ImageNet dataset.

4.2 Results

The algorithm is evaluated for 3 different amounts of corruption rate - 25% (Figure 2), 50% (Figure 3) and 100% (Figure 4).

Notation used in the graphs: ASL denotes the performance of the pure ASL, Co-ASL1 denotes the performance of the first DNN used in Co-ASL and Co-ASL2 denotes the performance of the second one. Each of the runs is for 80 epochs. The evaluation metric used is mAP score.

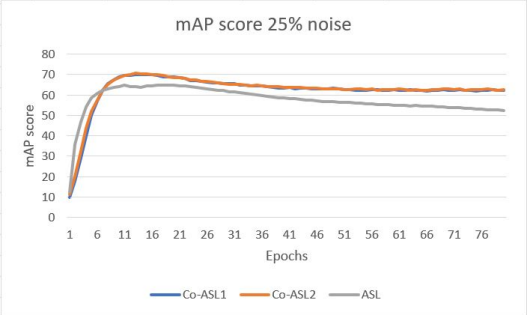


Figure 2: mAP score for 25% noise

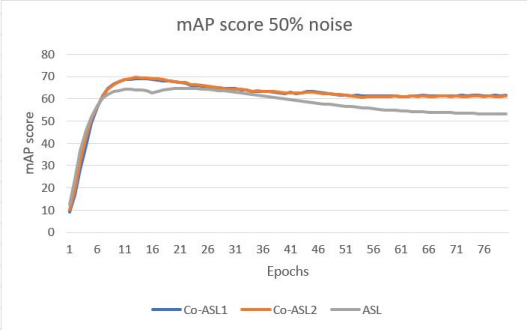


Figure 3: mAP score for 50% noise

General look

The behavior of both ASL and Co-ASL is quite similar for the cases of 25% and 50% - the 50% noise graph looks like the 25% one but just translated downward with a bit. ASL starts faster and achieves higher scores for the first five epochs. However, after that, the mechanism of Co-ASL to deal with noise starts to play a role, resulting in it scoring higher for the remaining epochs.

The 100% noise graph has a more distinct look. There the values of ASL and Co-ASL stay close for the first 60 epochs (except the region between epoch 20 and 30), but around epoch 60, ASL reaches its **overfitting point**.

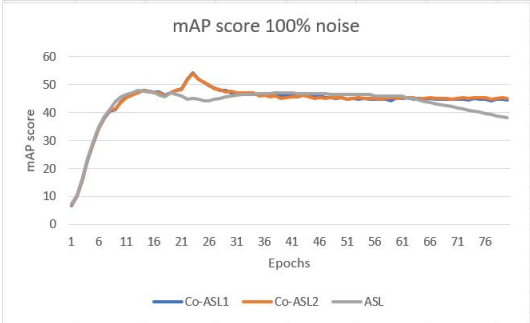


Figure 4: mAP score for 100% noise

Overfitting point

The overfitting point is the most crucial aspect in this research as it proves the improvement over the pure ASL. The overfitting point is the approximate epoch where ASL starts to overfit to the label corruption. As a result, the precision of the algorithm begins to drop. In contrast to ASL, Co-ASL doesn't have an overfitting point because of the robustness introduced by Co-teaching. In the example experiments, the overfitting point in the 25% and 50% noise cases is around epoch 30. In contrast, the point is reached around epoch 60 for the 100% corruption rate case. Despite the delay, the precision plummets faster in this case. As a result, the difference in the score at the last epoch is the biggest exactly in this case (Table 1).

$Method \backslash Noise$	25%	50%	100%
ASL	53.08	52.06	37.36
Co-ASL	61.54	60.58	46.12

Table 1: mAP score during the last epoch

5 Responsible Research

The focus of this section will be the ethical side of the research, and its two main aspects, research integrity and research reproducibility. During the research, there was no human-computer interaction. Also, as the publicly available dataset, MS-COCO was used as a data source, data collection was absent. Therefore, all the privacy-related issues are considered irrelevant, and the main accent will be the reproducibility of the conducted experiments.

The dataset used for training and evaluation, MS-COCO, is popularly used for MLL and is freely accessible online, and hence can be attained quite easily by anybody. How to inject noise into the dataset is explained in detail. However, because of the randomness used in the process, the resulting data may not always be the same, which could later lead to a slight deviation in the end results of the experiments.

The implementation of the proposed method is straightforward - the two modules on which it is based, Co-teaching and ASL, have their implementations published openly online. Their combination into Co-ASL is described with pseudo-code. The exact values of the hyper-parameters used by the model are also clearly stated.

6 Conclusions and Future Work

This paper introduced Co-ASL, a multi-label learning method robust to weak labels. It solves the problem by combining the high accuracy that ASL achieves on multi-label data with the robust training procedure introduced by Co-teaching. The experiments conducted showed an absence of overfitting even after 80 epochs and an improvement of between 8 and 9 mAP scores compared to the vanilla ASL.

Possible future improvements:

1. More in-depth evaluation.
    - Train on the whole MS-COCO dataset. The fact that the model was trained only on a third of it lowers the resulting accuracy. Also, some of the patterns of the data may not be exhibited because of that.
    - Evaluate against more types and amounts of noise - As the model was only tested against uniform symmetric noise and 3 different corruption rates, it may happen that the method is not robust for more sizeable and versatile noise.
    - Evaluate on other multi-label datasets, for example PASCAL VOC[6] and NUS-WIDE[5].
2. Try to extend ASL with some of the other multi-network approaches - Co-teaching+, Decoupling, and MentorNet and compare their performances.

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks, 2017.

[2] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification, 2021.

[3] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 92–100, New York, NY, USA, 1998. Association for Computing Machinery.

[4] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks, 2019.

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[7] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels, 2018.

[8] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, 2018.

[9] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data, 2017.

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[11] Wei Liu, Yu-Gang Jiang, Jiebo Luo, and S. Chang. Noise resistant graph ranking for improved web image search, 06 2011.

[12] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update", 2018.

[13] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling, 2019.

[14] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: a loss correction approach, 2017.

[15] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture, 2020.

[16] Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization, 2019.

[17] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey, 2021.

[18] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds., 01 2010.

[19] Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning with noisy label identification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6454–6461, Apr. 2020.

[20] Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification, 2020.

[21] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption?, 2019.

[22] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

[23] Min-Ling Zhang and Jun-Peng Fang. Partial multi-label learning via credible label elicitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[24] Wenting Zhao and Carla Gomes. Evaluating multi-label classifiers with noisy labels, 2021.