Learning to Detect Concepts from Webly-Labeled Video Data

Junwei Liang[1], **Lu Jiang**[1], **Deyu Meng**[2], **Alexander Hauptmann**[1]
[1]School of Computer Science, Carnegie Mellon University, PA, USA
[2]School of Mathematics and Statistics, Xi'an Jiaotong University, P. R. China.

Abstract

Learning detectors that can recognize concepts, such as people actions, objects, etc., in video content is an interesting but challenging problem. In this paper, we study the problem of automatically learning detectors from the big video data on the web without any additional manual annotations. The contextual information available on the web provides noisy labels to the video content. To leverage the noisy web labels, we propose a novel method called WEbly-Labeled Learning (WELL). It is established on two theories called curriculum learning and self-paced learning and exhibits useful properties that can be theoretically verified. We provide compelling insights on the latent non-convex robust loss that is being minimized on the noisy data. In addition, we propose two novel techniques that not only enable WELL to be applied to big data but also lead to more accurate results. The efficacy and the scalability of WELL have been extensively demonstrated on two public benchmarks, including the largest multimedia dataset and the largest manually-labeled video set. Experimental results show that WELL significantly outperforms the state-of-the-art methods. To the best of our knowledge, WELL achieves by far the best reported performance on these two webly-labeled big video datasets.

1 Introduction

The Internet has been witnessing an explosion of video data. Due to the huge volume of the data, automatic video understanding has received increasing attentions in both the artificial intelligence and the machine learning community. Generally, researchers are interested in training a large number of detectors that can automatically recognize concepts occurring in the video content, such as people, objects, actions, etc. These concept detectors are important building blocks for many applications such as video search, summarization and question answering [Jiang *et al.*, 2015b].

Training concept detectors on videos is more challenging than on still images. Manually labeling video requires playing back the video, which is more time consuming and expensive than labeling still images. As a result, the largest labeled video collection, called FCVID [Jiang *et al.*, 2015d], only contains about 0.09 million labels, much less than the 14 million labels in the image collection ImageNet [Deng *et al.*, 2009]. Furthermore, since videos are more complex than images, training robust video detectors require more labeled data. However, paradoxically, we have significantly less labels for videos than images where we should have more.

In fact, there exists considerable amount of videos on the web that contain rich contextual information with a weak annotation about the video content, such as the video title, description or the social network of the uploader. We call these videos webly-labeled. The webly-labeled videos can be collected without any manual effort, and its amount is orders of magnitude larger than that of any manually-labeled video collection. Unlike the manual labels, the web labels are noisy and have both low accuracy and low recall: the webly-labeled concepts may not present in the video content and concepts not in the web label may appear in the video.

Few studies have been proposed to leverage the noisy webly-labeled data in training concept detectors. Most of them are in the image domain [Fergus *et al.*, 2005; Li and Fei-Fei, 2010; Bergamo and Torresani, 2010]. For example, [Divvala *et al.*, 2014] proposed a semi-supervised learning method to extract concept variations and train image variation models based on downloaded images using text-based search. [Mitchell *et al.*, 2015] proposed a Never-Ending Language Learner that makes use of the Internet and learns knowledge and beliefs 24/7. A study by Google introduced an efficient large-scale video classification [Varadarajan *et al.*, 2015], where they utilized YouTube videos with weak web labels. These existing studies demonstrated promising results in this direction. However, existing methods are mainly built on heuristic approaches, and it is unclear, for example, what objective is being optimized; where or even whether the process converges. The lack of understanding of these questions hinders not only the theoretical analysis but also the practical advances of existing methods.

An ideal webly-labeled learning method would not only utilize heuristic but also, importantly, prove to be theoretically sound. To this end, this paper proposes a novel method called WEbly-Labeled Learning (WELL). It is established on the theories called *curriculum learning* [Bengio *et al.*, 2009] and *self-paced learning* [Kumar *et al.*, 2010]. The learning theory is inspired by the underlying cognitive processes

of humans and animals, which generally start with learning easier aspects of a task, and then gradually take more complex examples into consideration [Kumar *et al.*, 2010; Bengio *et al.*, 2009; Jiang *et al.*, 2015a]. Following their idea, WELL learns a concept detector iteratively from first using a few samples with more confident labels, then incorporates more samples with noisy labels. The algorithm combines the prior knowledge extracted from the webly-labeled data with the dynamic information learned from the statistical model to determine the label confidence in the next iteration.

WELL is a novel framework for training concept detectors from webly-labeled data. It is a also general framework that can incorporate state-of-the-art deep learning methods to learn robust detectors from noisy data that can also be applied to image domain. In summary, the contribution of this paper is threefold. First, it proposes a novel webly-labeled learning method with solid theoretical justifications. Second, it advances the state-of-the-art curriculum and self-paced theory by introducing two novel techniques, namely, the partial-order curriculum and dropout. The proposed techniques not only enable WELL to be applied to big data, but also lead to more accurate results. Finally, the efficacy and the scalability have been empirically demonstrated on two public benchmarks, including by far the largest manually-labeled video set called FCVID [Jiang *et al.*, 2015d] and the largest multimedia dataset called YFCC100M [Thomee *et al.*, 2015]. Experimental results show that WELL outperforms state-of-the-art methods with statistically significant differences. The promising results suggest that detectors trained on sufficient webly-labeled videos may outperform detectors trained on any existing manually-labeled sets.

2 Related Work

Curriculum and Self-paced Learning: Recently, Bengio et al. proposed a learning paradigm called *curriculum learning* (CL), in which a model is learned by gradually incorporating from easy to complex samples in training so as to increase the entropy of training samples [Bengio *et al.*, 2009]. A curriculum determines a sequence of training samples and is often derived by predetermined heuristics in particular problems. For example, [Chen and Gupta, 2015] designed a curriculum where images with clean backgrounds are ranked before the images with noisy backgrounds, i.e. their method first builds a feature representation by a Convolutional Neural Network (CNN) on images with clean background and then fine tunes the models on images with noisy background. In [Spitkovsky *et al.*, 2009], the authors approached grammar induction, where the curriculum is derived in terms of the length of a sentence. Because the number of possible solutions grows exponentially with the length of the sentence, and short sentences are easier and thus should be learn earlier.

The heuristic knowledge in a problem often proves to be useful. However, the curriculum design may lead to inconsistency between the fixed curriculum and the dynamically learned models. That is, the curriculum is predetermined prior knowledge and cannot be adjusted accordingly, taking into account the feedback about the learner. To alleviate the issue of CL, [Kumar *et al.*, 2010] designed a new paradigm, called *self-paced learning* (SPL). SPL embeds curriculum design as a regularizer into the learning objective. Compared with CL, SPL exhibits two advantages: first, it jointly optimizes the learning objective with the curriculum, and thus the curriculum and the learned model are consistent under the same optimization problem; second, the learning is controlled by a regularizer which is independent of the loss function in specific problems. This theory has been successfully applied to various applications, such as matrix factorization [Zhao *et al.*, 2015], action/event detection [Jiang *et al.*, 2014b], domain adaption [Tang *et al.*, 2012], tracking [Supancic and Ramanan, 2013] and segmentation [Kumar *et al.*, 2011], reranking [Jiang *et al.*, 2014a], etc.

Learning Detector in Web Data: Recently, a few studies have been proposed trying to utilize the huge amount of noisy data from the Internet. For example, [Mitchell *et al.*, 2015] proposed a Never-Ending Language Learning (NELL) paradigm and built adaptive learners that makes use of the web data by learning different types of knowledge and beliefs continuously. Such learning process is mostly self-supervised, and previously learned knowledge enables learning further types of knowledge. [Sukhbaatar *et al.*, 2014] designed loss layers specifically for noisy label learning of images in Convolutional Neural Network. It tried to estimate the distribution of noise and was mainly verified on synthesized noisy labels. [Liang *et al.*, 2015] presented a weakly-supervised method called Baby Learning for object detection from a few training images and videos. They first embed the prior knowledge into a pre-trained CNN. When given very few samples for a new concept, a simple detector is constructed to discover much more training instances from the online weakly labeled videos. As more training samples are selected, the concept detector keeps refining until a mature detector is formed. [Varadarajan *et al.*, 2015] discussed a method that exploits the YouTube topic API to train large scale video concept detectors on YouTube. The method utilized a calibration process and hard negative mining to train a second order mixture of experts model in order to discover correlations within the labels. Existing methods are mainly built on heuristic approaches and it is unclear what objective is being optimized. In this paper, we theoretically justify the proposed method and empirically demonstrate its superior performance over representative existing methods.

3 WEbly-Labeled Learning (WELL)

3.1 Model and Algorithm

In this paper, we consider a concept detector as a binary classifier. The noisy web labels for a concept can be automatically collected by matching the concept name to the latent topic of video metadata. For example, a video may have a web label "dog" as its title talks about dog. [Varadarajan *et al.*, 2015] utilizes the YouTube topic API, which is derived from the textual metadata, to automatically get noisy labels for videos. The web labels are quite noisy as the webly-labeled concepts may not present in the video content whereas the concepts not in the web label may well appear.

To leverage the noisy web labels in a principled way, we propose WEbly-Labeled Learning (WELL). Formally, given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^m$ denotes the feature for the i^{th} observed sample, and y_i represents its noisy

web label. Let $L(y_i, g(\mathbf{x}_i, \mathbf{w}))$, or ℓ_i for short, denote the loss function which calculates the cost between the noisy label y_i and the estimated label $g(\mathbf{x}_i, \mathbf{w})$. Here $\mathbf{w}$ represents the model parameter inside the decision function g. For example, in our paper, $\mathbf{w}$ represents the weight parameters in the Convolutional Neural Network (CNN). Our goal is to jointly learn the model parameter $\mathbf{w}$ and the latent weight variable $\mathbf{v} = [v_1, \cdots, v_n]^T$ by:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v};\lambda,\Psi) = \sum_{i=1}^{n} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda), \quad \text{subject to } \mathbf{v} \in \Psi \tag{1}$$

where $\mathbf{v} = [v_1, v_2, \cdots, v_n]^T$ denote the latent weight variables reflecting the labels' confidence. The weights determine a learning sequence of samples, where samples with greater weights tend to learned earlier. Our goal is to assign greater weights to the sample with confident labels whereas smaller or zero weights to the samples with noisy labels. To this end, we employ the self-paced regularizer f, which controls the learning scheme. We consider the binary regularizer Eq. (2) proposed in [Kumar *et al.*, 2010] and the linear regularizer Eq. (3) proposed in [Jiang *et al.*, 2015a]:

$$f_b(\mathbf{v};\lambda) = -\lambda\|\mathbf{v}\|_1, \tag{2}$$

$$f_l(\mathbf{v};\lambda) = \frac{1}{2}\lambda\sum_{i=1}^{n}(v_i^2 - 2v_i). \tag{3}$$

Generally, a self-paced regularizer determines the scheme for penalizing the latent weight variables. Physically it resembles the learning schemes human used in understanding new concepts. The linear scheme corresponds to a prudent strategy, which linearly penalizes the samples that are different to what the model has already learned (see Eq. (4)); whereas the binary scheme is more aggressive and only assigns binary weights. The hyper-parameter λ ($\lambda > 0$) is called "model age", which controls the pace at which the model learns new samples. When λ is small only samples of with small loss will be considered. As λ grows, more samples with larger loss will be gradually appended to train a "mature" mode.

Ψ in Eq. (1) is a curriculum region that incorporates the prior knowledge extracted from the webly-labeled data as a convex feasible region for the weight variables. The shape of the region weakly implies a prior learning sequence of samples, where the expected values for favored samples are larger. The curriculum region can be derived in a variety of ways. Section 7 will discuss this topic in details. A straightforward approach is by counting the term frequency in the video metadata. That is, for example, the chance of a video containing the concept "zebra" become higher when it has more word "zebra" in its title or description.

Eq. (1) represents a concise and general optimization model [Jiang *et al.*, 2015a]. It combines the prior knowledge extracted from the noisy webly-labeled data (as the curriculum region) and the information dynamically learned during the training (via the self-paced regularizer). Intuitively, the prior knowledge serves as an instructor providing a guidance on learning the latent weights, but it leaves certain freedom for the model (the student) to adjust the actual weights according to its learning pace. Experimental results in Section 4 demonstrate the learning paradigm can better overcome the noisy labels than just using either predetermined prior knowledge or dynamically learned information.

Algorithm 1: Webly-labeled Learning (WELL).

input : Input dataset $\mathcal{D}$, curriculum region Ψ, self-paced function f and a step size μ
output: Model parameter $\mathbf{w}$

1 Initialize $\mathbf{v}^*$, λ in the curriculum region;
2 **while** *not converged* **do**
3 Update $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{v}^*; \lambda, \Psi)$;
4 Update $\mathbf{v}^* = \arg\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}^*, \mathbf{v}; \lambda, \Psi)$;
5 **if** λ *is small* **then** increase λ by the step size μ;
6 **end**
7 **return** $\mathbf{w}^*$

Following [Kumar *et al.*, 2010; Jiang *et al.*, 2015a], we employ the alternative convex search algorithm to solve Eq. (1). Algorithm 1 takes the input of a curriculum region, an instantiated self-paced regulizer and a step size parameter; it outputs an optimal model parameter $\mathbf{w}$. First of all, it initializes the latent weight variables in the feasible region. Then it alternates between two steps until it finally converges: Step 4 learns the optimal model parameter with the fixed and most recent $\mathbf{v}^*$; Step 5 learns the optimal weight variables with the fixed $\mathbf{w}^*$. In the beginning, the model "age" is gradually increased so that more noisy samples will be gradually incorporated in the training. Step 4 can be conveniently implemented by existing off-the-shelf supervised learning methods such as the back propagation. Gradient-based methods can be used to solve the convex optimization problem in Step 5. According to [Gorski *et al.*, 2007], the alternative search in Algorithm 1 converges as the objective function is monotonically decreasing and is bounded from below.

At an early age when λ is small, Step 4 in Algorithm 1 has an evident suppressing effect over noisy samples that have greater loss to the already learned model. For example, with a fixed $\mathbf{w}$, the unconstrained close-formed solution for the regularizer in Eq. (3) equals

$$v_i^* = \begin{cases} -\frac{1}{\lambda}\ell_i + 1 & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda \end{cases}, \tag{4}$$

where v_i represents the ith element in the optimal solution $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]^T$. Eq. (4) called linear regularizer indicates the latent weight is proportional to the negative sample loss, and the sample whose loss is greater or equals to λ will have zero weights and thus will not affect the training of the next model. As the model age grows, the hyper-parameter λ increases, and more noisy samples will be used into training. The prior knowledge embedded in the curriculum region Ψ is useful as it suggests a learning sequence of samples for the "immature" model. Section 3.2 theoretically indicates that the iterative learning process is identical to optimizing a robust loss function on the noisy data.

If we keep increasing λ, the model will ultimately use every sample in the noisy data, which is undesirable as the labels of some noisy samples are bound to be incorrect. To this end, , we stop increasing the age λ after about a certain

number of iterations (early stopping). The exact stopping iteration for each detector is automatically tuned in terms of its performance on a small validation set.

Partial-order Curriculum

Ψ is a feasible region that embeds the prior knowledge extracted from the webly-labeled data. It physically corresponds to a convex search region for the latent weight variable. Given a set of training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, Jiang et al. proposed an implementation of the total-order curriculum on training samples [Jiang *et al.*, 2015a]. It is defined as a ranking function: $\gamma : \{\mathbf{x}_i\}_{i=1}^n \to \{1, 2, \cdots, n\}$, where $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$ represents that the sample x_i should be learned earlier than x_j in training. However, predetermining a total-order learning sequence for every pair of samples, especially in the big noisy data, seems to be infeasible. In reality, we can only obtain incomplete prior information from the noisy data. For examples, we may know videos with certain keywords in its title should be learned earlier, but may never know the learning priority for the videos that do not have the keywords.

To this end, we propose a novel notion called partial-order curriculum, which allows for leveraging the incomplete prior information residing in the webly-labeled data. Define a partial order relation $\preceq$ such that $x_i \preceq x_j$ indicates that the sample x_i should be learned no later than x_j ($i, j \in [1, n]$). Similarly given two sample subsets $\mathbf{X}_a \preceq \mathbf{X}_b$ denotes the samples in $\mathbf{X}_a$ should be learned no later than the samples in $\mathbf{X}_b$.

Definition 1 (Partial-order Curriculum) *Given the training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and their weight variables $\mathbf{v} = [v_1, \cdots, v_n]^T$. Define a partial-order set $\gamma = (\mathbf{X}, \preceq)$. For every element in set $\mathbf{X}_p \preceq \mathbf{X}_q (\mathbf{X}_p, \mathbf{X}_q \subseteq \mathbf{X})$, a feasible region $\Psi = (\mathbf{A}^T\mathbf{v} \leq 0)$ is called a partial-order curriculum region of γ if $\mathbf{A} = 0$ except $\forall x_i \in \mathbf{X}_p$, $\forall x_j \in \mathbf{X}_q$ we have $\exists t$, $\mathbf{A}_{ti} = -1$ and $\mathbf{A}_{tj} = 1$.*

The partial-order curriculum in Definition 1 generalizes the total-order curriculum by incorporating the incomplete prior over groups of samples. Samples in the confident groups should be learned earlier than samples in the less confident groups. It imposes no prior over the samples within the same group nor the samples not in any group. Definition 1 follows the curriculum definition in [Jiang *et al.*, 2015a] and will degenerate to the curriculum in [Jiang *et al.*, 2015a] when the partial order becomes the full order relation.

In our problem, we extract the partial-order curriculum in the following way: we only distinguish the training order for groups of samples. We directly utilize the textual descriptions of the videos generated by the uploaders. For each video, we extract the latent topics of the video based on their titles, descriptions and tags in their metadata. In terms of the distance between the video's latent topic to the target concept, we group videos in a sequential order for each concept. The grouping and ordering information of the videos can be used to construct the partial-order curriculum. In our experiment, we divide the data into two partial-order curriculum groups, where the videos with matching scores larger than zero are in one group and the rest are in the other group.

Dropout

The labels in webly-labeled data are much noisier than manually-labeled data, and as a result, we found that the learning is prone to overfitting the noisy labels. To address this issue, inspired by the dropout technique in deep learning [Srivastava *et al.*, 2014], we propose a dropout strategy for webly-labeled learning. It is implemented in the self-paced regularizer discussed in Section 3.1. With the dropout, the regularizers become:

$$\begin{aligned} r_i(p) &\sim \text{Bernoulli}(p) + \epsilon, (0 < \epsilon \ll 1) \\ f_b(\mathbf{v}; \lambda, p) &= -\lambda \|\mathbf{r} \cdot \mathbf{v}\|_1, \\ f_l(\mathbf{v}; \lambda, p) &= \frac{1}{2}\lambda \sum_{i=1}^{n} (\frac{1}{r_i} v_i^2 - 2v_i), \end{aligned} \tag{5}$$

where $\mathbf{r}$ is a column vector of independent Bernoulli random variables with the probability p of being 1. Each of the element equals the addition of r_i and a small positive constant ϵ. Denote $\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^n v_i \ell_i + f(\mathbf{v}; \lambda)$ as the objective with the fixed model parameters $\mathbf{w}$ without any constraint, and the optimal solution $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]^T = \arg\min_{\mathbf{v} \in [0,1]^n} \mathbb{E}_{\mathbf{w}}$. We have:

$$\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^{n} (\ell_i - r_i\lambda) v_i; \Rightarrow v_i^* = \begin{cases} 1 & \ell_i < r_i\lambda \\ 0 & \ell_i \geq r_i\lambda \end{cases}, \tag{6}$$

$$\begin{aligned} \mathbb{E}_{\mathbf{w}} &= \sum_{i=1}^{n} \ell_i v_i + \lambda (\frac{1}{2r_i} v_i^2 - v_i); \\ \frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} &= \ell_i + \lambda v_i / r_i - \lambda = 0; \\ \Rightarrow v_i^* &= \begin{cases} r_i(-\frac{1}{\lambda}\ell_i + 1) & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda \end{cases}. \end{aligned} \tag{7}$$

The dropout effect can be demonstrated in the closed-form solutions in Eq. (6) and Eq. (7): with the probability $1-p$, v_i^* in both the equations approaches 0; with the probability p, v_i^* approaches the solution of the plain regularizer discussed in Eq. (2) and Eq. (3). Recall the self-paced regularizer defines a scheme for learning samples. Eq. (6) and Eq. (7) represent the new dropout learning scheme.

When the base learner is neural networks, the proposed dropout can be used combined with the classical dropout in [Srivastava *et al.*, 2014]. The term dropout in this paper refers to dropping out samples in the iterative learning. By dropping out a sample, we drop out its update to the model parameter, which resembles the classical dropout used in neural networks. It operates on a more coarse-level which is useful for noisy data. When samples with incorrect noisy labels update a model, it will encourage the model to select more noisy labels. The dropout strategy prevents overfitting to noisy labels. It provides a way of combining many different sample subsets in different iterations in order to help avoid bad local minima. Experimental results substantiate this argument. In practice, we recommend setting two Bernoulli parameters for positive and negative samples on imbalanced data. Empirically, we apply a much smaller probability p on the negative samples than on the positive samples.

3.2 Theoretical Discussions

Interestingly, it turns out that Algorithm 1 actually optimizes an underlying non-convex robust loss on the noisy data. To

show this, let $v^*(\lambda, \ell)$ represent the optimal weight of v for a loss term ℓ imposed on a training sample in Eq (1), where

$$v^*(\lambda, \ell) = \text{argmin}_{v \in [0,1]} \ v\ell + f(v, \lambda). \tag{8}$$

For convenience of notation, let the curriculum region be the full space. According to [Meng and Zhao, 2015], the latent objective has the form of $\mathbb{E}_\ell = \sum_{i=1}^n F_\lambda(\ell_i)(\lambda > 0)$ with a latent loss function $F_\lambda(\ell)$ obtained by integrating the loss variable from $v^*(\lambda, \ell)$, i.e.,

$$F_\lambda(\ell) = \int_0^\ell v^*(\lambda; l) dl. \tag{9}$$

Note that in the above ℓ and l means loss variables in the latent loss function $F_\lambda(\ell)$ and the optimal weight function $v^*(\lambda, l)$, whereas ℓ_i denotes the loss value actually calculated on the i-th sample. Incorporate the binary and linear self-paced regularizers in Eq. (9), the latent objective becomes:

$$F_\lambda^b(\ell) = \min(\ell, \lambda) \tag{10}$$

$$F_\lambda^l(\ell) = \mathbf{I}(\ell \geq \lambda)\frac{\lambda}{2} + \mathbf{I}(\ell < \lambda)(\ell - \frac{\ell^2}{2\lambda}) \tag{11}$$

Eq. (10) and Eq. (11) are two common non-convex regularized penalties in the machine learning community, where Eq. (10) is the Capped-Norm based Penalty(CNP) [Zhang, 2010b; Gong *et al.*, 2013] and Eq. (11) is the Minimax Convex Plus (MCP) [Zhang, 2010a]. It has been showed that both CNP and MCP can be used as robust loss functions that threshold the samples of greater loss [Friedman *et al.*, 2007]. Therefore, Algorithm 1 actually minimizes a non-convex robust loss derived from the original loss in the base learner (e.g. hinge loss). On clean data, the effect of the robust loss may not be evident, but on noisy data, without the robust loss, the model can be easily dominated by a few noisy samples or outliers. Experimental results substantiate this argument, where we observed that the robust loss leads to more accurate results than the original loss on the webly-labeled data.

Based on this understanding, the proposed WELL can be theoretically justified from two independent perspectives. From the learning perspective, WELL mimics the human and animal learning process that learns a model gradually from confident to less confident examples in the noisy data. From the optimization perspective, on the other hand, it minimizes a non-convex robust loss (CNP or MCP) on the noisy data. The robust loss tends to depress samples with noisy labels or outliers. Due to the nature of non-convexity, WELL utilizes the curriculum and self-paced learning, which have been demonstrated to be instrumental in avoiding bad local minima in non-convex problems [Bengio *et al.*, 2009; Kumar *et al.*, 2010]. Interestingly, Meng and Zhao proved that when λ is fixed, Algorithm 1, in fact, is identical to the Majorization-Minimization algorithm [Mairal, 2013], a popular solver for non-convex problems [Meng and Zhao, 2015]. Based on the understanding, one can justify the role of the curriculum region, i.e. the curriculum confines the search space of a non-convex problem to some reasonable subspace which tends to improve the quality of the starting value and the final solution. The dropout methods on the other hand, prevent overfitting in the non-convex optimization problem.

4 Experiments

4.1 Experimental Setup

This section systematically verifies the accuracy and the scalability of the proposed method on learning concept detectors from noisy webly labeled video data. The experiments are conducted on two public benchmarks: FCVID and YFCC100M, where FCVID is by far one of the biggest manually annotated video set, and the YFCC100M dataset is the largest multimedia benchmark.

Dataset and Feature Fudan-columbia Video Dataset (FCVID) contains 91,223 YouTube videos (4,232 hours) from 239 categories. It covers a wide range of concepts like activities, objects, scenes, sports, etc. [Jiang *et al.*, 2015d]. Each video is manually labeled to one or more categories. In our experiments, we do not use the manual labels in training, but instead we automatically generate the web labels according to the concept name appearance in the video metadata. The manual labels are used only in testing to evaluate our and baseline methods. Following [Jiang *et al.*, 2015d], the standard train/test split and the same static CNN feature from [Jiang *et al.*, 2015d] are used to have a fair comparison to existing methods. The second set YFCC100M [Thomee *et al.*, 2015] contains about 800,000 videos on Flickr with metadata such as the title, tags, the uploader, etc. There are no manual labels on this set and we automatically generate the web labels from the metadata. We use the features provided in [Jiang *et al.*, 2015c] where we first extract the keyframe level the VGG neural network features [Chatfield *et al.*, 2014] and create a video feature by average pooling. The same features are used across different methods on each dataset. Since there are no annotations, we train the concept detectors on the most 101 frequent latent topics in the video metadata.

Baselines The proposed method is compared against the following five baseline methods which cover both the classical and the recent representative learning algorithms on webly-labeled data. *BatchTrain* trains a single SVM model using all samples with noisy labels. *AdaBoost* is a classical ensemble approach that combines the sequentially trained base classifiers in a weighted fashion [Friedman, 2002]. *Self-Paced Learning (SPL)* is a classical method where the curriculum is generated by the learner itself [Kumar *et al.*, 2010]. *BabyLearning* is a recent method that simulates baby learning by starting with few training samples and fine-tuning using more weakly labeled videos crawled from the search engine [Liang *et al.*, 2015]. We build a search engine that indexes the textual metadata and retrieves videos using concept words based on Lucene [Białecki *et al.*, 2012]. *GoogleHNM* We use the hard negative mining strategy in [Varadarajan *et al.*, 2015]. On FCVID, we use the YouTube topic API to acquire the noisy label whereas on YFCC100M we obtain the noisy label by the Lucene search engine.

Evaluation Metrics On FCVID, as the manual labels are available, the performance is evaluated in terms of the precision of the top 5 and 10 ranked videos (P@5 and P@10) and mean Average Precision (mAP) of 239 concepts. On YFCC100M, since there are no manual labels, for evaluation, we apply the detectors to a third public video collection called TRECVID MED which includes 32,000 Internet

videos [Over *et al.*, 2014]. We apply the detectors trained on YFCC100M to the TRECVID videos and manually annotate the top 10 detected videos of each method for 101 concepts.

Our Model We build our method on top of a pre-trained convolutional neural network as the low-level features, i.e. static CNN features on FCVID and VGG features on YFCC100M. The concept detectors are trained based on a hinge loss cost function. Algorithm 1 is used to train the concept models iteratively and the λ stops increasing after 100 iterations. We automatically generate noisy web labels based on the video metadata. For the videos with noisy positive labels, we group them based on their latent topics, and derive a partial-order curriculum in Definition 1. The hyper-parameters of all methods including the baseline methods are tuned on the same validation set. On FCVID, the set is a small training subset with manual labels whereas on YFCC100M it is a proportion of noisy training set.

4.2 Experiments on FCVID

Table 1 compares the precision and mAP of different methods where the best results are highlighted. As we see, the proposed WELL with dropout significantly outperforms all baseline methods, with a significant difference at p-level of 0.05. For example, WELL outperforms the best baseline on 194 out of 239 concepts. The promising experimental results substantiate our theoretical analysis in Section 3.2. With the proposed model, the binary and linear regularizer yield a similar accuracy on this dataset. The performance difference between WELL with and without dropout demonstrates the efficacy of the proposed dropout technique, and the difference between SPL and WELL indicates the benefit of incorporating the proposed partial-order curriculum.

Note, WELL does not use any manual labels in training, but interestingly, its accuracy is comparable with the model trained on 35,850 videos with ground truth labels in [Jiang *et al.*, 2015d]. To investigate the potential of training concepts on webly-labeled video data, we apply WELL on the data subsets of different sizes. Specifically, we randomly split the FCVID training set into the subset of 200, 500, 1,000, and 2,000 hours of videos, and train the models on each subset. The models are then tested on the same standard test set. Table 2 lists the results. As we see, the accuracy of WELL on webly-labeled data increases along with the growth of the size of noisy data. The accuracy of WELL on 2,000 hours of videos with noisy web labels turns out to be better than the model trained on 500 hour of manually labeled data. Recall FCVID is one of the biggest manually annotated set which contains about 2,000 hours of annotated videos. According to the results, we hypothesize that with more webly-labeled data, which is not hard to obtain, WELL can potentially outperform models trained on any existing manually-labeled data.

4.3 Experiments on YFCC100M

Since there are no manual labels on YFCC100M, to evaluate the performance, we manually annotate the top 10 videos in the test set and report their precisions in Table 3. A similar pattern can be observed where the comparisons substantiate the rationality of the proposed partial-order curriculum and the dropout technique. The promising results on the largest multimedia set YFCC100M verify the scalability of the proposed method.

Table 1: Performance comparison on FCVID.

Method	P@5	P@10	mAP
BatchTrain	0.782	0.763	0.469
Adaboost [Friedman, 2002]	0.456	0.412	0.293
SPL [Kumar *et al.*, 2011]	0.793	0.754	0.414
GoogleHNM [Varadarajan *et al.*, 2015]	0.781	0.757	0.472
BabyLearning [Liang *et al.*, 2015]	0.834	0.817	0.496
WELL (binary w/o dropout)	0.857	0.843	0.521
WELL (linear)	**0.893**	**0.877**	**0.566**
WELL (binary)	**0.893**	**0.878**	**0.567**

Table 2: MAP comparison of models trained using web labels and ground-truth labels on different subsets of FCVID.

Dataset Size	200h	500h	1000h	2000h
WELL (noisy web label)	0.413	0.480	0.520	0.567
BatchTrain (ground-truth label)	0.485	0.561	0.604	0.638

Table 3: Performance comparison on YFCC100M.

Method	P@3	P@5	P@10
BatchTrain	0.535	0.513	0.487
Adaboost [Friedman, 2002]	0.341	0.327	0.282
SPL [Kumar *et al.*, 2011]	0.485	0.463	0.454
GoogleHNM [Varadarajan *et al.*, 2015]	0.541	0.525	0.500
BabyLearning [Liang *et al.*, 2015]	0.548	0.519	0.466
WELL (binary w/o dropout)	0.607	0.608	0.589
WELL (linear)	**0.667**	**0.663**	**0.649**
WELL (binary)	**0.660**	**0.640**	**0.625**

5 Conclusions

In this paper, we proposed a novel method called WELL for webly labeled video data learning. WELL extracts informative knowledge from noisy weakly labeled video data from the web through a general framework with solid theoretical justifications. It further improves curriculum and self-paced learning theory with the partial-order curriculum and dropout to build better video detectors with noisy data. WELL achieves the best performance only using webly-labeled data on two major video datasets. The result suggests that with more webly-labeled data, which is not hard to obtain, WELL can potentially outperform models trained on any existing manually-labeled data.

Acknowledgments

This work was partially supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. It was also based in part on work supported by the National Science Foundation (NSF) under grant number IIS-1251187. Deyu Meng was partially supported by the NSFC project (61373114). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[Bergamo and Torresani, 2010] Alessandro Bergamo and Lorenzo Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[Białecki *et al.*, 2012] Andrzej Białecki, Robert Muir, and Grant Ingersoll. Apache lucene 4. 2012.

[Chatfield *et al.*, 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[Chen and Gupta, 2015] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Divvala *et al.*, 2014] Santosh K Divvala, Alireza Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[Fergus *et al.*, 2005] Robert Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning object categories from google's image search. In *ICCV*, 2005.

[Friedman *et al.*, 2007] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[Friedman, 2002] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[Gong *et al.*, 2013] Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Z Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, 2013.

[Gorski *et al.*, 2007] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[Jiang *et al.*, 2014a] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *MM*, 2014.

[Jiang *et al.*, 2014b] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *NIPS*, 2014.

[Jiang *et al.*, 2015a] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *AAAI*, 2015.

[Jiang *et al.*, 2015b] Lu Jiang, Shoou-I Yu, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *ICMR*, 2015.

[Jiang *et al.*, 2015c] Lu Jiang, Shoou-I Yu, Deyu Meng, Yi Yang, Teruko Mitamura, and Alexander G Hauptmann. Fast and accurate content-based semantic search in 100m internet videos. In *MM*, 2015.

[Jiang *et al.*, 2015d] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[Kumar *et al.*, 2010] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, 2010.

[Kumar *et al.*, 2011] M Pawan Kumar, Haithem Turki, Dan Preston, and Daphne Koller. Learning specific-class segmentation from diverse data. In *ICCV*, 2011.

[Li and Fei-Fei, 2010] Li-Jia Li and Li Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International journal of computer vision*, 88(2):147–168, 2010.

[Liang *et al.*, 2015] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *ICCV*, 2015.

[Mairal, 2013] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *NIPS*, 2013.

[Meng and Zhao, 2015] Deyu Meng and Qian Zhao. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.

[Mitchell *et al.*, 2015] T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, et al. Never-ending learning. In *AAAI*, 2015.

[Over *et al.*, 2014] Paul Over, Jon Fiscus, Greg Sanders, David Joy, Martial Michel, George Awad, Alan Smeaton, Wessel Kraaij, and Georges Quénot. Trecvid 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2014.

[Spitkovsky *et al.*, 2009] Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Baby steps: How less is more in unsupervised dependency parsing. *NIPS GRLL*, 2009.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[Sukhbaatar *et al.*, 2014] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

[Supancic and Ramanan, 2013] James Steven Supancic and Deva Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.

[Tang *et al.*, 2012] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, 2012.

[Thomee *et al.*, 2015] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[Varadarajan *et al.*, 2015] Balakrishnan Varadarajan, George Toderici, Sudheendra Vijayanarasimhan, and Apostol Natsev. Efficient large scale video classification. *arXiv preprint arXiv:1505.06250*, 2015.

[Zhang, 2010a] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.

[Zhang, 2010b] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *JMLR*, 11:1081–1107, 2010.

[Zhao *et al.*, 2015] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *AAAI*, 2015.