Active Learning for Structured Prediction from Partially Labeled Data

Anonymous ICCV submission

Paper ID 843

1. Training details

Learning Rate and Weight Decay Policy: Our strategy involves continually fine-tuning a network given iterations of new labeled data. Choosing the right learning rate is an important consideration. At every iteration of our active learning algorithm, the solver starts with a base learning rate and decreases it gradually. Initializing the solver with the same learning rate at the beginning would result in low performance since the model has already been fine-tuned using some data. So in our policy, at each iteration, the solver is automatically initialized with a base learning rate that is smaller than the previous iteration’s base learning rate by a fixed factor. In order to avoid over-fitting to the smaller amount of data in the beginning, we designed a weight decay policy that the solver at first is initialized with a larger weight decay and at each iteration it decreases by a constant factor. In our experiments we used 0.5 for the base learning rate multiplication factor and 0.1 for weight decay.

2. Explanation of the first stage

The first stage (learner/classifier) of our active learning method could be any structured prediction learning algorithm that involves inference on a graphical model. For the implementation, we used the work of Deng et al. [1] as the learner/classifier. The reason is two fold: (1) it is state of the art; (2) the code is publicly available and is a good fit for our problem. For better understanding of our code/method, here we briefly explain [1].

Deng et al. [1] develop a method for structured prediction within deep networks. The approach implements the message passing algorithm for graphical model inference on a deep neural network framework. Their framework has two parts (see Fig. 1). The first consists of individual predictions for each output of the model. In the case of activity recognition, these would correspond to predicting the action label for each person as well as the overarching activity label for the scene. Convolutional neural networks (AlexNet [2]) are used for these individual predictions. With two different types of output node (action / scene), two different neural networks are trained, A-CNN an S-CNN. These two networks will be used later for extracting features from the frame and bounding boxes of the people in the scene.

The second part is an inference machine that uses the outputs from these individual neural networks to refine estimates of labels. This second part is akin to a message passing algorithm to conduct inference in a graphical model. Parameters in these messages control how much support / conflict there is between different labels in a scene, similar to potential functions in a graphical model. These parameters are represented as weights in a neural network and can be learned by back-propagation.

Network parameters. The parameters that we used for A-CNN and S-CNN of the first stage of our algorithm are as follow. We set the initial base learning rate of A-CNN and S-CNN to 0.0005 and 0.00005 respectively; and we set the initial weight decay of both to 0.05. In the experiment section we explained that we start with large base learning rate and weight decay and gradually reduce it at each iteration. Based on our heuristic approach, we multiply the base learning rate by 0.2 at each iteration. Similar to learning rate, we multiply the weight decay by 0.2 at each iteration. Then the model is initialized with the new parameters and learning continues. The batch sizes that we used for A-CNN and S-CNN are 600 and 384 respectively. As for the second stage we set the initial base learning rate to 0.002 and at every five iteration we multiply it by 0.5. However, for second stage we used the same weight decay and multiplier that we used for A-CNN and S-CNN.

2.1. Experiments

We have conducted two sets of experiments on two different datasets. The plots in Fig. 4 of the paper illustrates all the comparison of our method to the baselines on Collective Activity Dataset (Table 1a) and Volleyball Dataset (Table 1b). These plots are based on the numbers in the Table 1. The tables on the left and on the right show the results of adding $K = 1000$ and $K = 500$ annotations per iteration, respectively. The first column of the tables shows the accuracy of the trained model on the initial training set, which is same for all the methods. Subsequent columns show the results of the iterations of active learning.
Table 1: Results of comparison of our method against baselines on a) Collective Activity Dataset and b) The Volleyball Dataset. The numbers in the table are accuracies of the action and scene labels(%). For all the methods we start from same small initial labeled set so the accuracies of the first column are exactly the same. For each dataset, two sets of experiments are conducted that differ in number of annotations added at each iteration. Tables on the left and on the right report results of experiments with 1000 and 500 number of annotations added per iteration, respectively. For all the experiments, models are trained for 60 number of epochs at each iteration.

**References**


[6] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *Com-
Figure 1: Overview of the Stage 1 learner. Bounding boxes of individuals and the entire frame are passed through two CNNs to predict their labels. This information is then passed to a Recurrent Neural Network (called BP-RNN). The BP-RNN simulates the belief propagation algorithm on the graphical model of the scene, in which there are nodes corresponding to the overarching scene label and action label of every person.