CS 7150: Deep Learning — Summer-Full 2020 — Paul Hand

Week 8 — Preparation Questions For Class
Due: Monday June 29, 2020 at 12:00 PM Eastern time via Gradescope

Name: Saurabh Vaidya
Collaborators: Sumeet Gajjar

You may consult any and all resources in answering the questions. Your goal is to have answers that are ready to be shared with the class (or on a hypothetical job interview) as written. **Make sure to tag each question when you submit to Gradescope.**

Directions: Read the article 'Learning From Noisy Singly-labeled Data'.

Question 1. *Provide a summary of the contributions of this paper.*

Response:

Answer 1. *This paper contributed a new approach to learning true labels of data while also incorporating a predictor. They present MBEM where we can get good estimates of worker quality i.e annotators while also getting an optimal trained predictor. Both the predictor and worker quality help each other in estimating their better values such that the likelihood of observed labels is maximized.*

The paper shows us that we can make effective use for our annotation budget to learn an optimal predictor by employing only one label per to be collected per example and increasing a lot of annotated examples provided that worker quality is above some threshold.

Question 2. *What model for rater errors do the authors assume? What is unreasonable about this model?*

Response:

Answer 2. *About the error rate of a worker the authors assume that probability of error of a worker labeling a class label k as label s is independent of the input data .i.e it is constant for a pair of $s - k$ for that worker.*

Lets set the context of image classification with input image and output a class label. The authors assume that probability of a worker labelling a true label of, for example a dog as cat, is not dependent on what input image they were looking at.

*This is unreasonable since the decision of a worker to label a dog as cat would actually be influenced by how much confusing that **particular input image** was for the worker. Its unreasonable to assume the worker has a constant(uniformly random) probability of labelling a dog as cat. It is very well dependent on the input.*

Question 3. *The authors introduce a Model Bootstrapped EM (MBEM) algorithm. They compare their method to the EM algorithm and a weighted-EM algorithm. Clearly present the EM and weighted-EM algorithms. Succinctly state the difference between MBEM and EM.*

Response:

Answer 3. *In EM algorithm, we first estimate true labels with, for example,majority voting and using those estimates of ground truth we compute the worker quality. We use the new estimates of worker quality to compute updated estimates of true labels. We keep repeating this so that it maximizes the likelihood estimation of observed labels. We keep repeating this process until our estimates of labels and worker quality converge. When EM converges we get a posterior distribution* $P_{\hat{\pi},\hat{q}}(Y_i = k|Z^r, w^r)$ *on the labels.*

In weighted-EM, we first run EM to get final estimates of worker quality(π*) and estimates of ground truth i.e. we get posterior distribution of labels. We then train a model with a weighted loss function where the loss function value for a label s is multiplied by its posterior probability, for all* $s \in \mathcal{K}$ *. When the model is learned we get our final estimates of true labels as the predictions of the model. Hence in weighted-EM, the learning is done* ***after*** *EM has converged with posteriors as weights.*

The difference between MBEM and EM is that in EM we iteratively compute our true labels and worker qualities + class priors alternatively until they converge. The final converged posterior distribution of labels given worker's observed labels gives us true label estimates. Using these estimates of true label we can proceed with our learning problem. In MBEM, the estimates of worker quality and class label priors are used to compute the posterior probabilities of estimated true labels. However, instead of just using the posteriors as estimates of true labels we use them to learn a loss function. The predictor on this loss function is our source of true label estimates. The learned predictor's output is the used as estimates of true labels.

However, the predictor is part of our alternating EM steps. We compute worker quality estimates using model predictions as true labels and keep alternating betweeen updates of worker qualities and true labels until they converge. The end output is a learned predictor function and estimate of worker confusion matrix.

Question 4. *Does the MBEM algorithm involve directly training a neural network? If so, which line of Algorithm 1 involves training a net?*

Response:

Answer 4. *The MBEM does involve directly training a model(neural network). The line where we learn predictor function i.e*

$$\widehat{f} \leftarrow \underset{f \in \mathcal{F}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \sum_{k \in K} \mathbb{P}_{\widehat{\pi},\widehat{q}}\Big[Y_i = k|Z_i^r; w_i^r\Big] \ell(f(X_i), Y_i = k) \tag{1}$$

is the training step. This step gives us a trained predictor whose predictions would be used as estimates of true labels.

Question 5. *If only one label is provided for a given training image, how is it even possible to assess the quality of that training label? What information is being leveraged that makes this possible?*

Response:

Answer 5. *It is possible to assess the quality of a training label if only one label is provided, if we know the probability of error of the worker who labelled that training input. If that worker has labelled sufficiently large number of examples, we can compute the probability of error of that worker.*

Authors leverage the trained model as the ground truth to evaluate worker quality.i.e the confusion matrix π^a_{ks}. Having the trained model's output as true label, simply the workers that agree more with the model's output are better than those who disagree with the model a lot.

Question 6. *In Figure 1, why is the 'Oracle weighted EM' curve shown? Why is the 'Oracle correctly labeled' curve shown? Would it be possible for any algorithm to beat 'Oracle correctly labeled'? If not, why does the dashed line on the bottom of the graph show superior performance?*

Response:

Answer 6. *The 'Oracle weighted EM' model is trained on loss function with true confusion matrix. MBEM learns a weighted loss function with weights estimate from estimated worker confusion matrix. The oracle weighted EM curve is shown to compare how closely MBEM performs implying how accurately MBEM predicts the true confusion matrix.*

The 'Oracle correctly labeled' model is trained on standard loss function but only on those examples where atleast one worker gave the true label. Thus a model trained on these labels has eliminated training on those examples for which we did not have a true label at all. Thus it will have a better generalization error since it did not learn any noisy/incorrect labels. MBEM when plotted against this curve, can show us how robust MBEM is to mislabeling by all workers. i.e how well it performed inspite of learning on some noisy labels or how much effect did the noisy labels had on MBEM performance.

Oracle correctly labeled model is a black box that knows true label from at least one worker, which means if may not get all examples from original training set. Any algorithm will always get the entire training set, including those examples where no worker gave a true label. Thus any algorithm which succeeds in being agnostic to those noisy label examples will probably match the performance of oracle correctly labeled model. It can never beat it even if have all training examples where at least one worker gives the true label when the predictor is the same.

The dashed line is generalization error of model trained with ground truth values of labels on all training examples. Since we have all true labels the worker quality and redundancy has no effect on generalization error. Thus it will always have superior performance than oracle correctly labeled because we have trained on all example with true labels versus what oracle correctly labeled would use.

Question 7. *Explain Figure 1. Make sure to set up the context, state what is plotted, state what is to be observed, and state what conclusion is reached. What are the dashed lines?*

Response:

Answer 7. *The authors want to compare their algorithm against baseline models(weighted-MV, weighted-EM, MBEM) and oracle models on CIFAR-10 dataset. To compare their algorithm they have a hammer-spammer style worker skill distribution where a hammer worker will label true label with some probability γ and spammer worker choose label uniformly at random. They compare these styles in class-wise manner too where worker would be hammer for subset of classes where the confusion matrix value is 1 at diagonals and 0 elsewhere. For other subset of classes worker would be a spammer with matrix value of $\frac{1}{}$*

The observations and conclusions are same in both styles i.e hammer-spammer and class wise.

Figure 1 has three plots in 2 scenarios of hammer-spammer and class wise hammer spammer. The first plot shows generalization errors of different models vs worker quality with redundancy=1.It is observed that with increasing worker quality, performance of all algorithms improves. MBEM outperforms all *baseline models for all worker quality values and matches closely with oracle weighted EM. It is concluded that MBEM is able to accurately model the worker confusion matrix to true confusion matrix of workers.*

Plot 2 shows generalization error versus redundancy where the probability of a hammer worker $\gamma = 0.2$*. It is observed that weighted EM and weighted MV perform better than EM and MV respectively. It implies that using weighted loss functions with weights as posterior probabilities is effective. It is also seen that at small redundancy, MBEM performs much better than weighted-EM implying that, bootstrapping model training inside EM is effective. At large redundancies the EM and MBEM work equally good*

Plot 3 also plots generalization error vs redundancy with $\gamma = 0.2$ *but the annotation budget is fixed. This would mean if we have more redundancy we would label less examples and if we want to label more examples we will get less redundancy. It is observed that as number of redundancy increases, the generalization error also increases for MBEM due to less examples for training. Weighted EM performs better when redundancy increases from 1 to 5. This implies that with standard EM its effective to collect less examples with more redundancy as EM has better worker quality estimations. MBEM has the lowest error of all when the redundancy is 1. This verifies their theoretical result that single label annotation is optimal when worker quality is above some threshold.*

The dashed lines show the generalization error of a model that is trained with ground truth labels on all training examples. This model since it has all true labels does not get affected by changing the worker quality or redundancy.