Confident Learning: Estimating Uncertainty in Dataset Labels

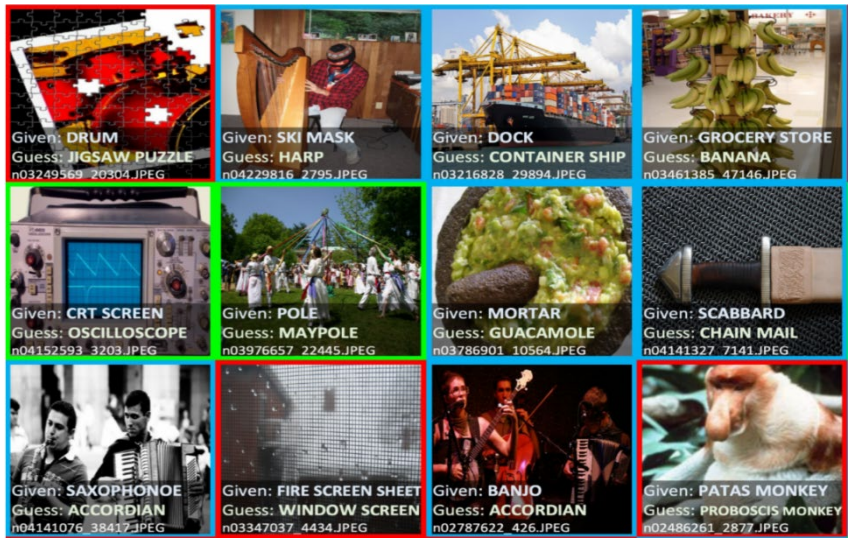
2020.6.24

CurtisG.Northcutt LuJiang IsaacL.Chuang

Content

motivation

There is no completely clean data set



Multi-label image(blue)

Ontological problem(green)

label error(red)

How to identify and clean noisy label ?

M.I.T. and Google researchers tried to clean up the data in what they called "Confident Learning," in a semi-supervised way.

How does Confident Learning Work?

1. let's imagine we have a dataset with images of dogs, foxes, and cows.

2. Get a confidence matrix

out-of-sample predicted probabilities
matrix size: # of examples(400) by # of classes (3)

noisy labels
vector length: number of examples (400)

$C_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	100	40	20
$\tilde{y}=fox$	56	60	0
$\tilde{y}=cow$	32	12	80

$$C_{\tilde{y},y^*}[i][j] := |\hat{X}_{\tilde{y}=i,y^*=j}| \quad \text{where}$$

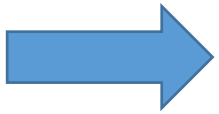
$$\hat{X}_{\tilde{y}=i,y^*=j} := \left\{ x \in X_{\tilde{y}=i} : \hat{p}(\tilde{y}=j;x,\theta) \geq t_j,\ j = \underset{k \in M: \hat{p}(\tilde{y}=k;x,\theta) \geq t_k}{\arg\max} \hat{p}(\tilde{y}=k;x,\theta) \right\}$$

input $\hat{P}$ an $n \times m$ matrix of out-of-sample predicted probabilities $\hat{P}[i][j] := \hat{p}(\tilde{y}=j;x,\theta)$
input $\tilde{y} \in \mathbb{N}_{\geq 0}{}^n$, an $n \times 1$ array of noisy labels
procedure CONFIDENTJOINT($\hat{P}$, $\tilde{y}$):
PART 1 (COMPUTE THRESHOLDS)
for $j \leftarrow 1, m$ **do**
  for $i \leftarrow 1, n$ **do**
    $l \leftarrow$ new empty list []
    if $\tilde{y}[i] = j$ **then**
      append $\hat{P}[i][j]$ to l
  $t[j] \leftarrow$ average(l)
PART 2 (COMPUTE CONFIDENT JOINT)
$C \leftarrow m \times m$ matrix of zeros
for $i \leftarrow 1, m$ **do**
  $cnt \leftarrow 0$
  for $j \leftarrow 1, m$ **do**
    if $\hat{P}[i][j] \geq t[j]$ **then**
      $cnt \leftarrow cnt + 1$
      $y^* \leftarrow j$
  $\tilde{y} \leftarrow \tilde{y}[i]$
  if $cnt > 1$ **then**
    $y^* \leftarrow \arg\max \hat{P}[i]$
  if $cnt > 0$ **then**
    $C[\tilde{y}][y^*] \leftarrow C[\tilde{y}][y^*] + 1$
output C $m \times m$ unnormalized counts matrix

How does Confident Learning Work?

3. Estimate the joint distribution of noisy and true labels and find label issues

$C_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	100	40	20
$\tilde{y}=fox$	56	60	0
$\tilde{y}=cow$	32	12	80



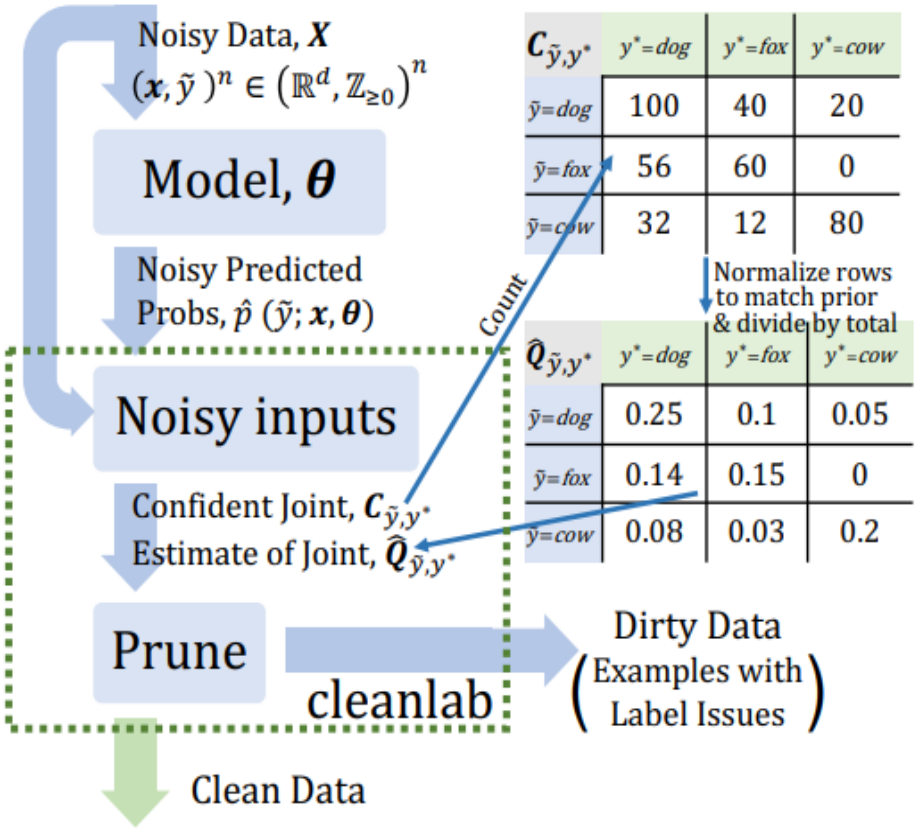
$\hat{Q}_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	0.25	0.1	0.05
$\tilde{y}=fox$	0.14	0.15	0
$\tilde{y}=cow$	0.08	0.03	0.2

4. Clean the data set

non-diagonal entries in the matrix marked as label issues

How does Confident Learning Work?

Framework



$\boldsymbol{C}_{\tilde{y},y^*}$	y^*=dog	y^*=fox	y^*=cow
$\tilde{y}$=dog	100	40	20
$\tilde{y}$=fox	56	60	0
$\tilde{y}$=cow	32	12	80

$\widehat{\boldsymbol{Q}}_{\tilde{y},y^*}$	y^*=dog	y^*=fox	y^*=cow
$\tilde{y}$=dog	0.25	0.1	0.05
$\tilde{y}$=fox	0.14	0.15	0
$\tilde{y}$=cow	0.08	0.03	0.2

Tow inputs

Same dataset
Different parameters or model

cross-validation

1. out-of-sample predicted probabilities (matrix size: # of examples by # of classes)

2. noisy labels (vector length: number of examples)

Three steps

1. Estimate the joint distribution.

2. Find and prune noisy examples with label issues.

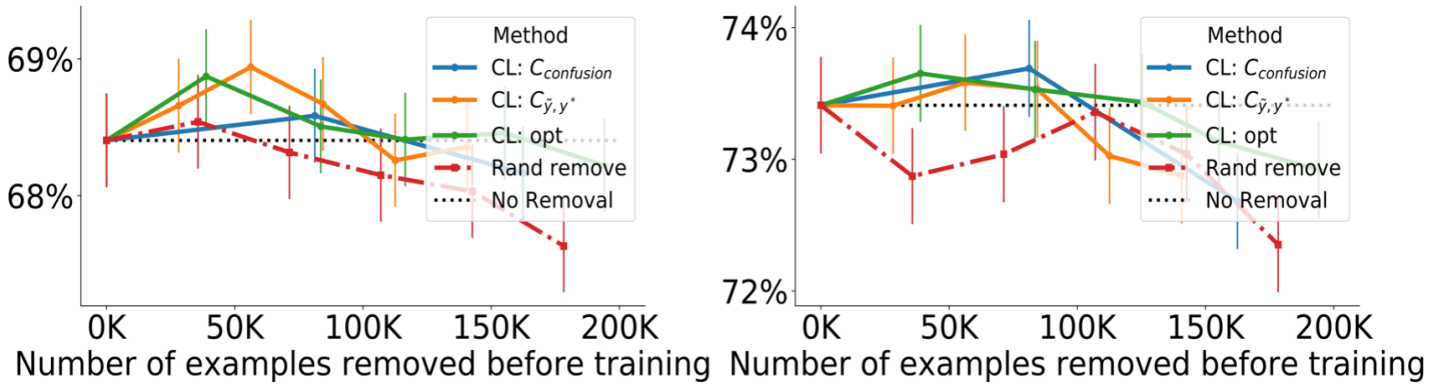
3. Train with errors removed.

Practical Applications of Confident Learning

Noise		0.2				0.4				0.7			
Sparsity	**AVG**	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $C_{\text{CONFUSION}}$	0.662	0.854	0.854	0.863	0.857	0.806	0.796	0.802	0.798	0.332	0.363	0.328	0.291
CL: $C_{\tilde{y},y^*}$	0.673	0.848	0.858	0.862	0.861	**0.815**	**0.810**	**0.816**	0.815	0.340	0.398	0.282	**0.372**
CL: OPT	**0.696**	0.860	0.859	0.865	**0.862**	0.810	0.801	0.814	**0.825**	**0.468**	**0.420**	**0.399**	0.371
SCE-LOSS	0.615	**0.872**	**0.875**	**0.888**	0.844	0.763	0.741	0.649	0.583	0.330	0.287	0.309	0.240
MIXUP	0.622	0.856	0.868	0.870	0.843	0.761	0.754	0.686	0.598	0.322	0.313	0.323	0.269
MENTORNET	0.590	0.849	0.851	0.832	0.834	0.644	0.642	0.624	0.615	0.300	0.316	0.293	0.279
CO-TEACHING	0.569	0.812	0.813	0.814	0.806	0.629	0.616	0.609	0.581	0.305	0.302	0.277	0.260
S-MODEL	0.556	0.800	0.800	0.797	0.791	0.586	0.612	0.591	0.575	0.284	0.285	0.279	0.273
REED	0.560	0.781	0.789	0.808	0.793	0.605	0.604	0.612	0.586	0.290	0.294	0.291	0.268
BASELINE	0.554	0.784	0.792	0.790	0.782	0.602	0.608	0.596	0.573	0.270	0.297	0.282	0.268

a comparison of CL versus recent state-of-the-art approaches for multiclass learning with noisy labels on CIFAR-10.

Training on ImageNet cleaned with CL Improves ResNet Test Accuracy



(a) ResNet18 Validation Accuracy

(b) ResNet50 Validation Accuracy

Final Thoughts

Benefits of confidence learning

confident learning requires no hyperparameters

We use cross-validation to obtain predicted probabilities out-of-sample

Compare CL with AL

Active learning is an process of incrementing a data set.

Confident learning is an process of cleaning a data set

Can we apply CL to task segmentation ?

The probability value of each pixel with non-zero predicted probability is used as the probability of the sample

Thanks