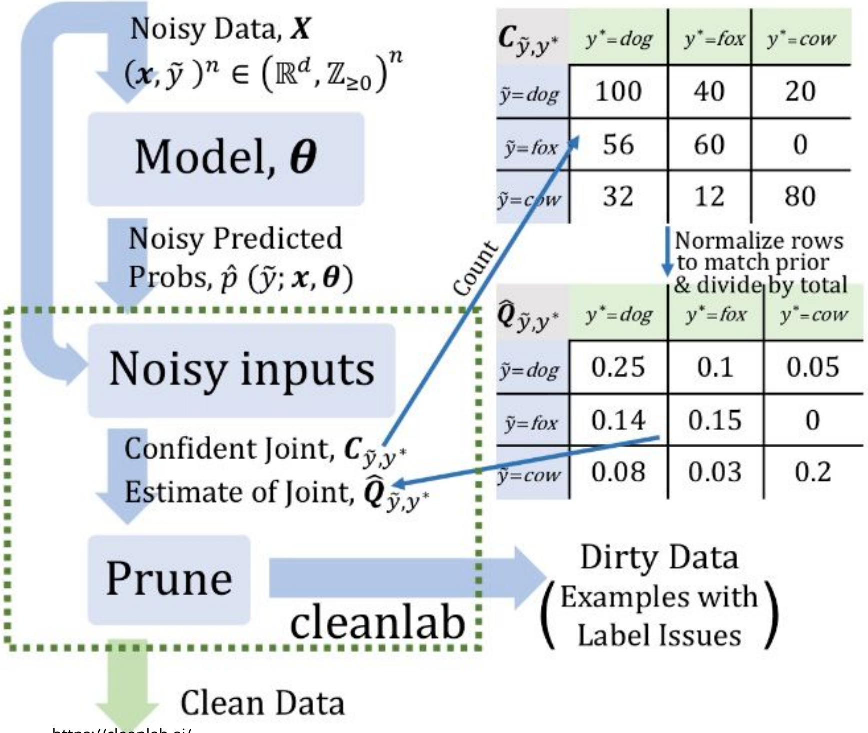
Confident Learning applied to MNIST Label Error

David Szczecina

Confident Learning and Cleanlab



Noisy Data, $\boldsymbol{X}$

$(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, \mathbb{Z}_{\geq 0})^n$

Model, $\boldsymbol{\theta}$

Noisy Predicted Probs, $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$

Noisy inputs

Confident Joint, $\boldsymbol{C}_{\tilde{y},y^*}$

Estimate of Joint, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$

Prune

cleanlab

Dirty Data (Examples with Label Issues)

Clean Data

Count

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

Normalize rows to match prior & divide by total

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

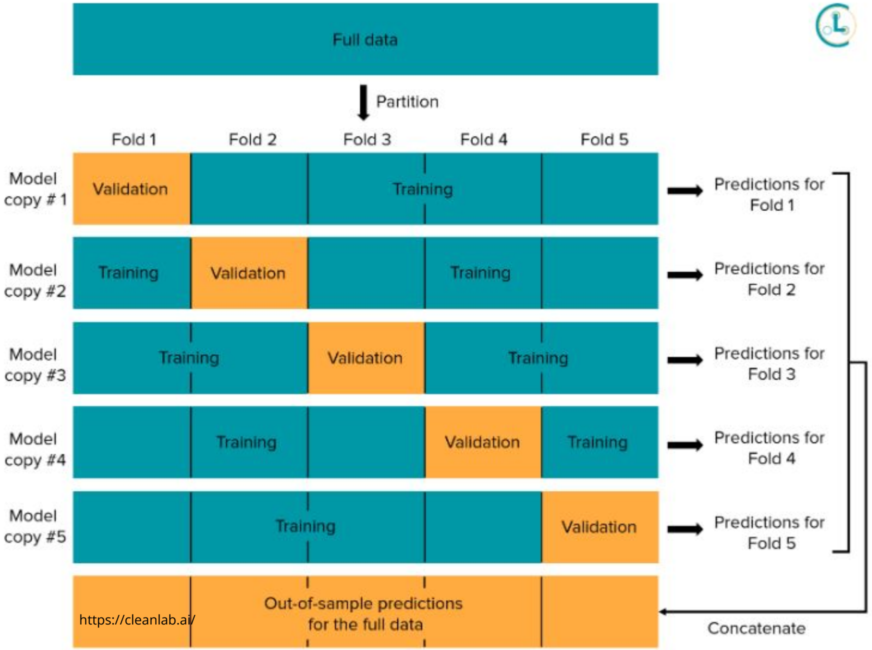
https://cleanlab.ai/

1. Estimate the joint distribution of given, noisy labels and latent (unknown) uncorrupted labels to fully characterize class-conditional label noise.
2. Find and prune noisy examples with label issues.
3. Train with errors removed, re-weighting examples by the estimated latent prior.

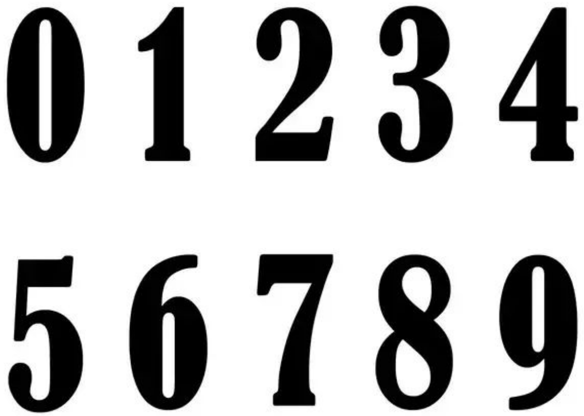
https://arxiv.org/pdf/1911.00068.pdf

Confident Learning and Cleanlab

Out of sample Predictions + Labels (can contain label errors)



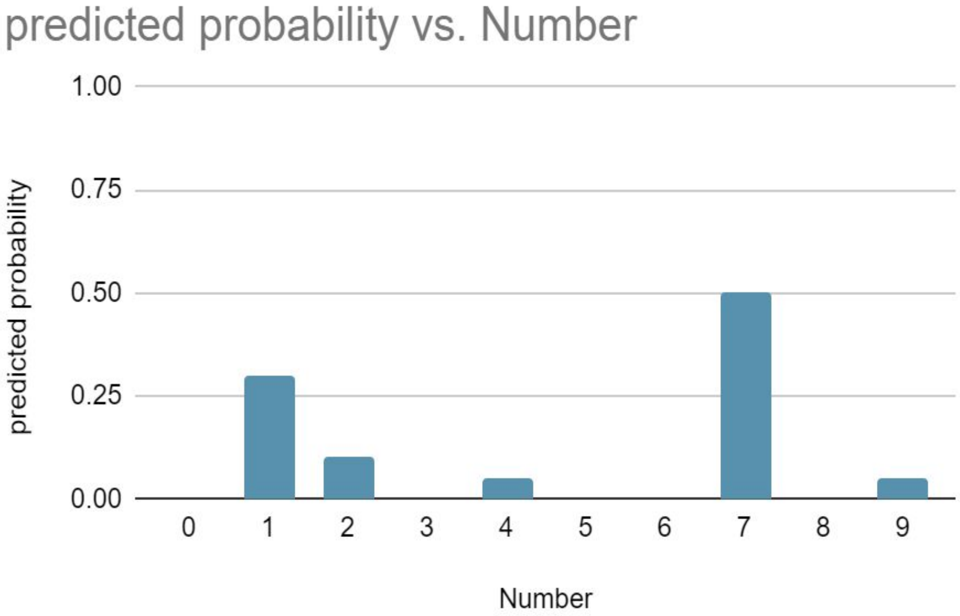
https://cleanlab.ai/



Predicted Probabilities

The central idea is that when the predicted probability of an example is greater than a per-class-threshold, we *confidently count* that example as actually belonging to that threshold's class.

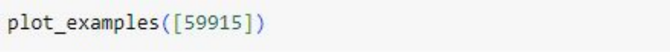
The thresholds for each class are the average predicted probability of examples in that class.



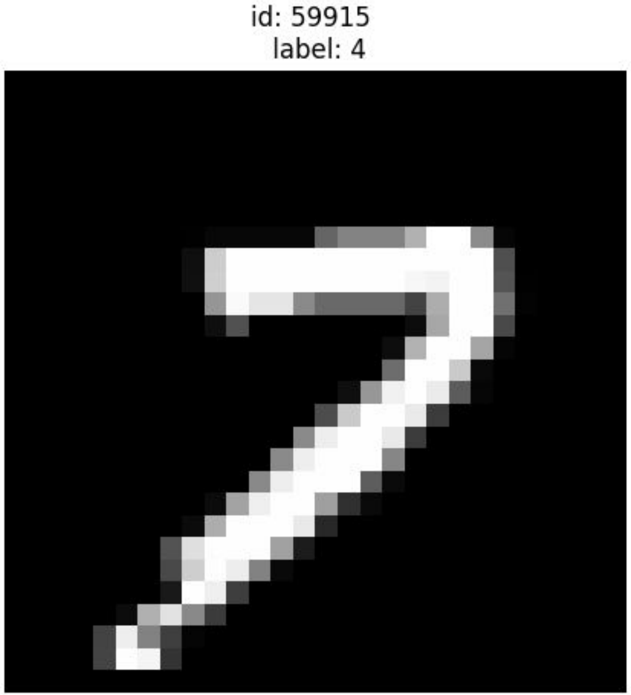
MNSIT Example

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

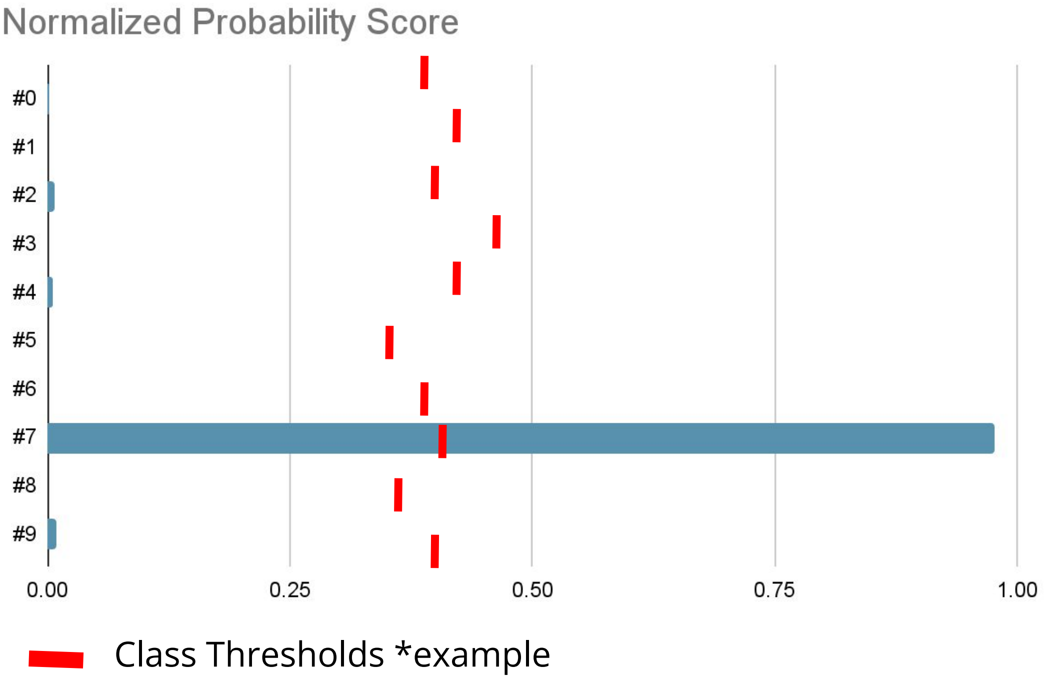
$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} := \left\{ \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j,\; j = \mathop{\arg\max}_{l \in [m] : \hat{p}(\tilde{y}=l;\boldsymbol{x},\boldsymbol{\theta}) \geq t_l} \hat{p}(\tilde{y}=l;\boldsymbol{x},\boldsymbol{\theta}) \right\}$$



```
plot_examples([59915])
```



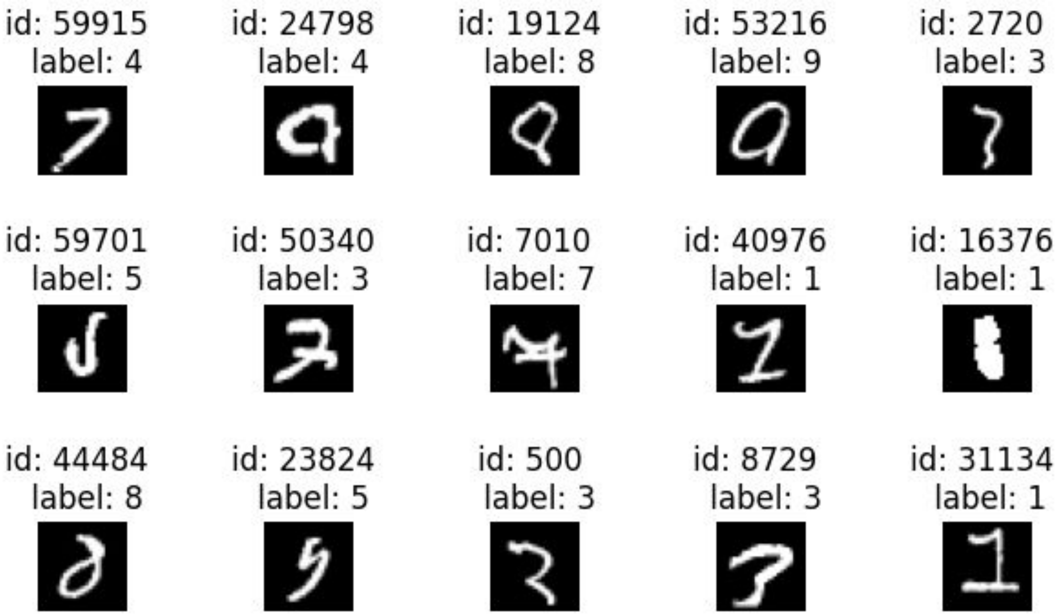
$$t_j = \frac{1}{|\boldsymbol{X}_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})$$



Results

148 out of 70000 labels identified as bad.

Many are mislabeled or questionable, some are correct and false positives.



Ranking Calculation

'normalized_margin': normalized margin (p(label = k) - max(p(label != k)))

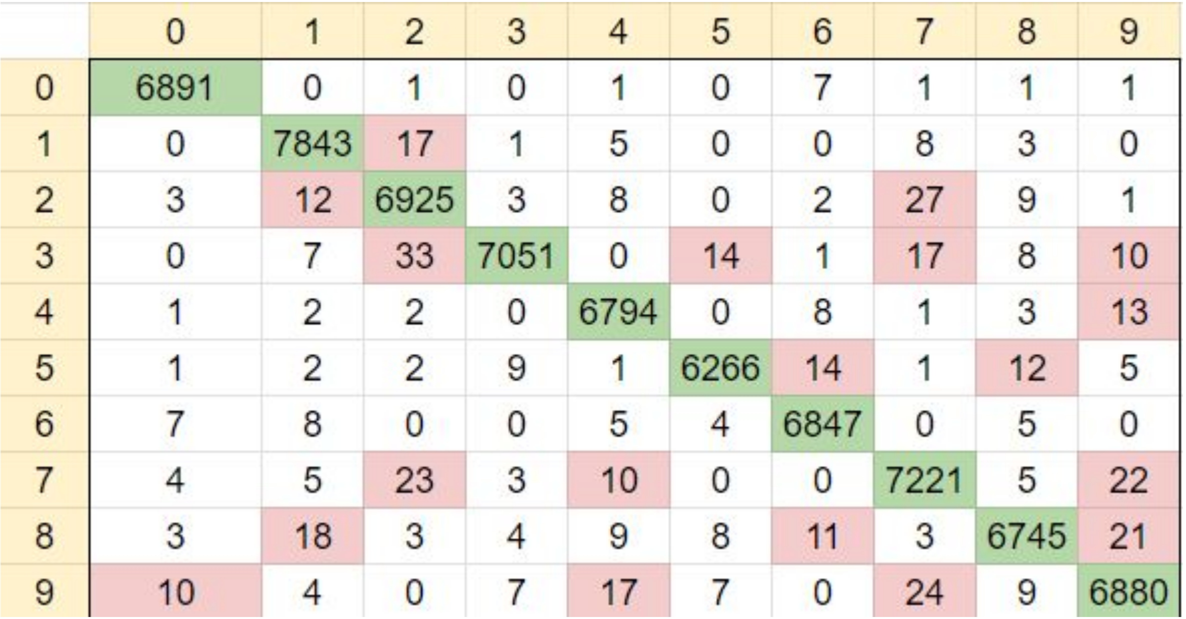
'self_confidence': [pred_probs[i][labels[i]] for i in label_issues_idx]

'confidence_weighted_entropy': entropy(pred_probs) / self_confidence

Confusion Matrix

Predicted

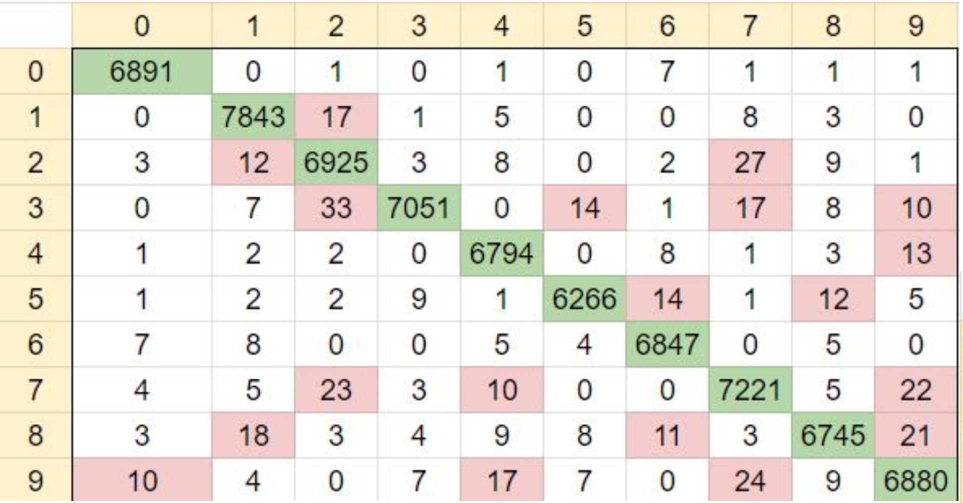
Given Label



	0	1	2	3	4	5	6	7	8	9
0	6891	0	1	0	1	0	7	1	1	1
1	0	7843	17	1	5	0	0	8	3	0
2	3	12	6925	3	8	0	2	27	9	1
3	0	7	33	7051	0	14	1	17	8	10
4	1	2	2	0	6794	0	8	1	3	13
5	1	2	2	9	1	6266	14	1	12	5
6	7	8	0	0	5	4	6847	0	5	0
7	4	5	23	3	10	0	0	7221	5	22
8	3	18	3	4	9	8	11	3	6745	21
9	10	4	0	7	17	7	0	24	9	6880

Total Errors: 148

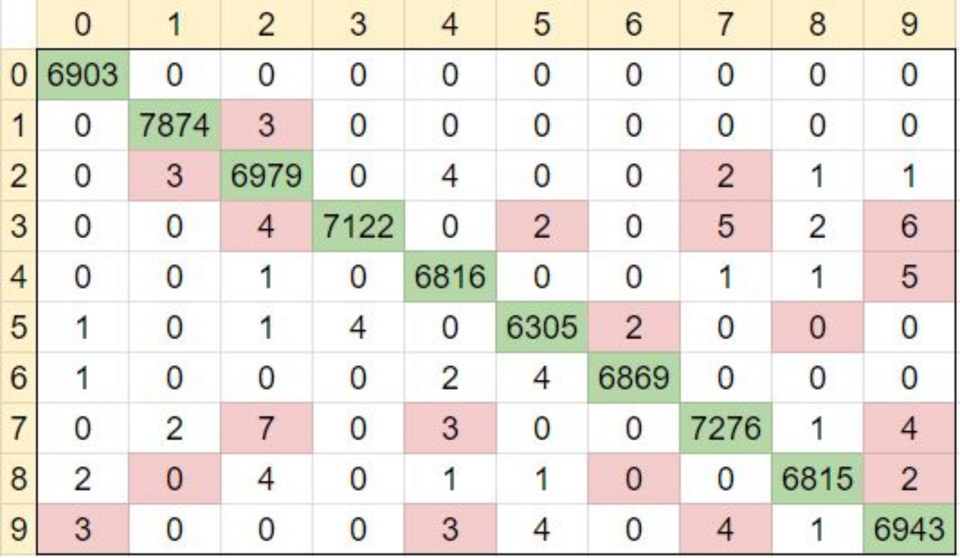
3 Folds vs 10 Folds



	0	1	2	3	4	5	6	7	8	9
0	6891	0	1	0	1	0	7	1	1	1
1	0	7843	17	1	5	0	0	8	3	0
2	3	12	6925	3	8	0	2	27	9	1
3	0	7	33	7051	0	14	1	17	8	10
4	1	2	2	0	6794	0	8	1	3	13
5	1	2	2	9	1	6266	14	1	12	5
6	7	8	0	0	5	4	6847	0	5	0
7	4	5	23	3	10	0	0	7221	5	22
8	3	18	3	4	9	8	11	3	6745	21
9	10	4	0	7	17	7	0	24	9	6880

3 fold cross validation: 148 errors

10 fold cross validation: 98 errors



	0	1	2	3	4	5	6	7	8	9
0	6903	0	0	0	0	0	0	0	0	0
1	0	7874	3	0	0	0	0	0	0	0
2	0	3	6979	0	4	0	0	2	1	1
3	0	0	4	7122	0	2	0	5	2	6
4	0	0	1	0	6816	0	0	1	1	5
5	1	0	1	4	0	6305	2	0	0	0
6	1	0	0	0	2	4	6869	0	0	0
7	0	2	7	0	3	0	0	7276	1	4
8	2	0	4	0	1	1	0	0	6815	2
9	3	0	0	0	3	4	0	4	1	6943

Next Steps

- See difference in model accuracy without bad labeled data
- See how well it works with additional augmented data
- Test on BIOSCAN-small