Spotting Spurious Data with Neural Networks

Hadi Amiri, Timothy A. Miller, Guergana Savova

Harvard Medical School, Boston, MA

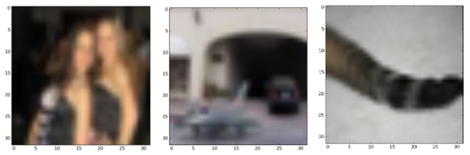
{firstname.lastname}@childrens.harvard.edu

Objectives

Automatic identification of spurious instances (those with potentially wrong labels in datasets). We aim to improve the quality of existing resources, especially when annotations are obtained through crowdsourcing or automatically generated based on coded rankings.

Introduction

Spurious instances can mislead systems, and, if available in test data, lead to unrealistic comparison among competing systems. Some samples:



(a) Truck (b) Airplane (c) Cat

Figure 1: Sample noise in CIFAR-10 dataset.

Story: Mary traveled to the garden. Daniel went to the garden. Mary journeyed to the kitchen. Mary went **back to the hallway**. Daniel traveled to the office. Daniel moved to the garden. Sandra went back to the kitchen. John traveled to the bathroom.

Question: Where is Mary?

Answer: hallway

Story: John went to the office. Daniel journeyed to the office. Sandra picked up the football **there**. Sandra went to the bedroom. Sandra left the football there. Sandra went **back to the kitchen**. Sandra traveled to the hallway. Sandra moved to the garden.

Question: Where is the football?

Answer: bedroom

Table 1: Sample inconsistencies in bAbi dataset.

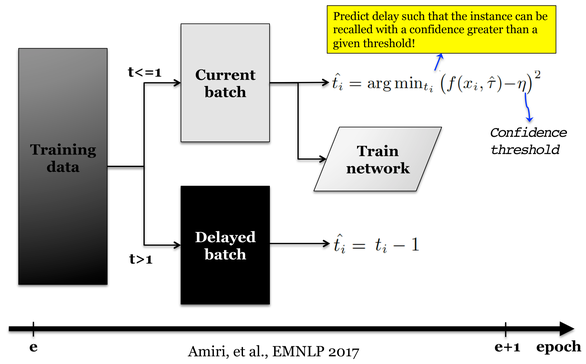
Contributions

- A cognitively-motivated and effective algorithm for identifying spurious instances in datasets,
- Our approach can be applied to *any* dataset without modification if there exists a neural network architecture for the target task of the dataset.

- Code: `scholar.harvard.edu/hadi/spot`
- `hadi.amiri@childrens.harvard.edu`

Developing Noise Spotters

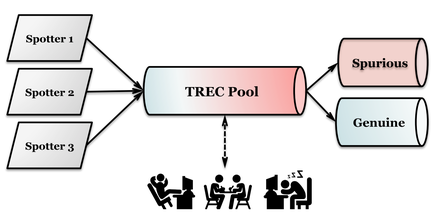
Spaced Repetition



Amiri, et al., EMNLP 2017

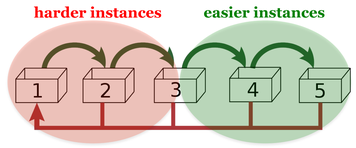
- Training instances are repeatedly presented to a learner on a schedule determined by a spaced repetition algorithm,
- Such algorithms are inspired by factors that affect human memory retention, namely, *difficulty* of learning materials, *delay* since their last review, and *strength* of memory,
- Scheduler increases intervals of time between subsequent "reviews" of previously learned materials,
- Efficient and effective training paradigms for neural networks (Amiri et al., EMNLP 2017).

TREC-based Evaluation



- Addition: 10K/2K, noise_level = (0,0.5]
- Twitter: 10K/1K, noise_level = 0.30
- Reddit: 4K/400, noise_level = 0.23

Leitner Spotter



$$S^e(h_i) = S^{e-1}(h_i) + \mathbb{1}_{q_0^e}(h_i) \times \left(\frac{1}{|q_0^e|} + loss^e(h_i)\right)$$

Spurious confidence of instance i at epoch e

Weighted loss of instance i if it occurs in first queue

Increase scores of instances that *frequently* occur in the first queue.

- Suppose we have n queues $\{q_0, q_1, \ldots, q_{n-1}\}$,
- Initially places all instances in the first queue, q_0,
- Leitner scheduler trains the network only with instances of q_i at every 2^i iterations,
- During training, if an instance from q_i is correctly classified by the network, the instance will be "promoted" to q_{i+1}, otherwise it will be "demoted" to the first queue, q_0,
- As network trains, higher queues will accumulate easier instances, while lower queues carry either hard or potentially spurious instances.

Algorithm 2. Leitner Spotter

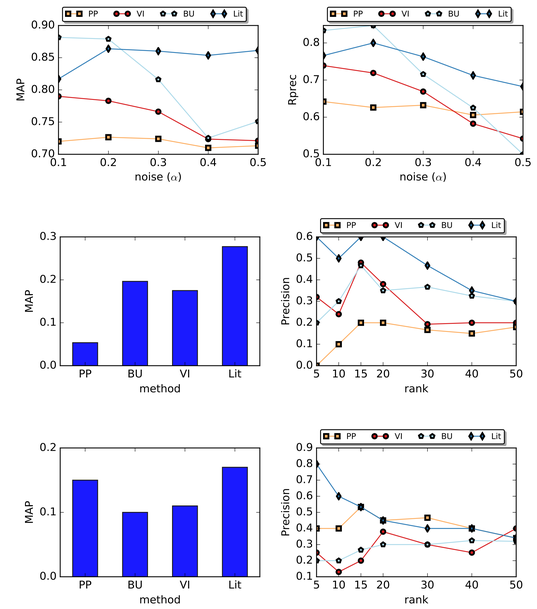
Input: $\mathbf{H}$: training data, $\mathbf{V}$: validation data, k : number of iterations, n : number of queues
Output: Ranked list of spurious instances

$Q = [q_0, q_1, \ldots, q_{n-1}]$
$q_0 = [\mathbf{H}]$, $q_i = []$ for $i \in [1, n-1]$
$S^0[h_j] = 0$ for $h_j \in \mathbf{H}$
For $epoch = 1$ to k:
  $batch = []$
  For $i = 0$ to $n-1$:
    If $epoch \% 2^i == 0$:
      $batch = batch + q_i$
  End For
  $promos, demos, loss = \mathbf{train}(batch, \mathbf{V})$
  $\mathbf{update\_queue}(Q, promos, demos)$
  $S^{epoch} = \mathbf{update\_stat}(S^{epoch-1}, Q, loss)$
End for
return $\mathbf{sort}(S^k, \mathbf{H}, loss)$

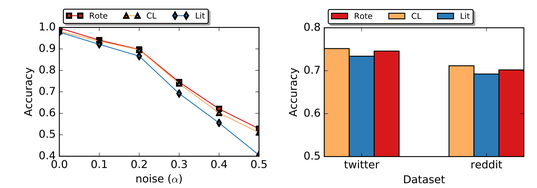
q_0 epochs = $\{1, 2, 3, 4, 5, \ldots\}$
q_1 epochs = $\{2, 4, 6, 8, 10, \ldots\}$
q_2 epochs = $\{4, 8, 12, 16, 20, \ldots\}$

Experiments

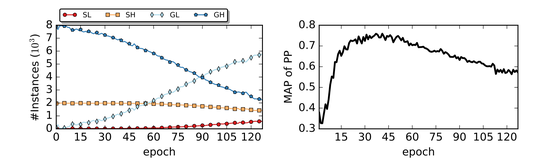
Noise Detection



Training Performance



Why `loss` alone is not enough?



(i) Loss classes (j) MAP of Loss spotter

Takeaways

1. Most instances do not need to be used at every epoch when training neural networks.
2. Spurious instances frequently occur in the first queue of Leitner system.
3. Loss alone is not enough to identify spurious instances.