On the value of modeling dependencies in weak supervision

Salva Rühling Cachay[1], Benedikt Boecking[2], Artur Dubrawski[2]
[1]Technical University of Darmstadt
[2]Carnegie Mellon University

1. Motivation

- Modern ML requires large amounts of labeled data
- Data labeling is expensive, both in terms of time & cost.

  → Promising alternative: *data programming*.

  Encode domain knowledge into *labeling functions (LFs)* that noisily label subsets of the data

- LFs are denoised by a label model, producing *probabilistic labels* used to train an end classifier.
- LFs often have *statistical dependencies* (Fig. 1).
- Specifying the correct dependencies is hard!
- *What happens if we misspecify the generative model?*

2. Contributions

- *Theoretically bound* the generalization risk under model misspecification.
- *Empirically* show that the error can be quite substantial, even when the *dependency structure appears sensible*.

3. Experiments & Results

- We select a) 135 LFs for an IMDB review dataset, and b) 85 LFs for a biography classification task. Model proxies for the unknown true dependencies.
- Modeling more than a few ***seemingly sensible*** dependencies (as in Table 1) ***deteriorates*** downstream test ***performance*** by up to 4 AUC points in a), 8 AUC points in b)

```
def labeling_function1(text_sample):
  return POSITIVE if "recommend" in text_sample else ABSTAIN
def labeling_function2(text_sample):
  return NEGATIVE if "wouldn't recommend" in text_sample else ABSTAIN
```

Fig 1.: Example LFs for a sentiment classification task. Potential dependency: LF2 *fixes* the less precise vote of LF1 when both label. Should we model it?

4. Discussion & Future Work

Does the observed loss in performance occur due to a) the true model having few dependencies; or b) the more complex model with dependencies being too sample inefficient?

5. Conclusions

Modeling dependencies can boost performance.

But seemingly sensible dependencies between labeling functions often have *marginal* or even *negative* effects on the downstream classifier.

Thus, *ignoring dependencies is often a reasonable baseline* for practitioners.

Table 1: Strongest and weakest dependencies (IMDB LFs)

LF_j	LF_k	value	type
worth	not worth	219	negated
special	not special	8	negated
original	bad	327	priority
recommend	terrible	53	priority
recommend	highly recommend	226	reinforcing
bad	absolutely horrible	7	reinforcing

$$\text{Generalization risk} \leq \gamma + ||\Delta\text{accuracy\_params}||_1 + ||\text{dependency\_params}||_1$$

of a misspecified label model

LF labeling **accuracy parameter estimation error** w.r.t. to the true model

Parameter excess from **incorrectly modeled** dependencies

OR

Magnitude of the true dependencies that **weren't modeled**

6. Acknowledgments

RISS ROBOTICS INSTITUTE SUMMER SCHOLARS
DAAD Deutscher Akademischer Austausch Dienst German Academic Exchange Service