Rejection learning for responsible automation  
a short thread (thread icon)  
<https://gph.is/g/aNMzDjv>

Many actionable prediction tasks are actually automation problems – we want to replace human effort with algorithms.   
We tend to equate prediction and automation, but this reduction may be unhelpful for solving the problem we care about.

Using a rejection mechanism is one aspect often neglected in the design of prediction models.   
When a model has the ability to abstain from making a prediction, the resulting system is expected to be a lot more sensible.

In healthcare, this would mean models that are safer, better performing, and trusted by physicians.  
Human effort is still needed to treat the rejected cases, but many times it’d be worth it. Especially when compared to the previous situation of 100% human effort.

See this work <https://arxiv.org/abs/1903.12220> by @maithra\_raghu, @greg\_corrado, @oziadias @m\_sendhil and this work and this work <https://www.nature.com/articles/s41746-020-00367-3> by @BenKompa, @AndrewLBeam, @latentjasper for an excellent introduction/motivation to the topic

Accuracy-rejection curves are powerful visualizations that can aid the design of classifiers with rejection.

<http://proceedings.mlr.press/v8/nadeem10a.html>

How can we learn a classification rule with rejection? A few approaches. First is the notion of uncertainty – the model should reject instances it is not confident about.

This is consistent with the optimal Bayes classifier:

Chow.png

Problem: accurate estimation of P(Y|X) is hard! Uncertainty quantification is an active area of research, and while I’m not familiar with it, AFAIK no uncertainty method managed to show guarantees w.r.t the multiclass rejection learning problem..

Second approach is to learn both a classifier h and a rejector r. The motivation is added flexibility. For linear models, only decoupling of classifier and rejector can solve this toy example (credit https://cs.nyu.edu/~mohri/pub/rej.pdf)  
flexibility.png

Many heuristics fit in this approach – for example, we can use an outlier detection algorithm to reject some instances, and learn a classifier independently. But again, no theoretical guarantees are available for multiclass classification.

A recent work by @NolfwinMk + that IMO has good potential proposed to learn a single model (like the confidence approach) but skips the need to accurately learn all P(Y|X): only two values are needed for the optimal classification rule – max and argmax (P(Y|X))

Chow.png

Using a reduction to cost sensitive learning, they present a theoretically sound objective function that can be easily learned:

<https://arxiv.org/abs/2010.11748>

Obj.png

Can we do better? It can be argued that in most automation problems, any decent ML will have very low error on a subset of the population, and vice versa – only a human expert could solve some fraction of the cases.

If we could predict not only which are the error prone cases, but rather which cases would benefit more from the algorithm and which from the human expert – the performance of the combined ‘system’ would surpass each of the ‘components’ alone.

This framing is sometimes termed ‘learning to defer’ and very interesting results have been shown by @david\_madras in <https://arxiv.org/abs/1711.06664>, by @HsseinMzannar @david\_sontag in <https://arxiv.org/abs/2006.01862> and also <https://arxiv.org/abs/1903.12220> mentioned before.

But the caveat is that two sets of labels are needed to learn these models – a ground truth and a decision maker’s predictions. I think that it is not a reasonable assumption in most applications, because only a single label is available for each instance.

These were a few words about rejection learning. My impression is that it’s a super useful framework that doesn’t get enough attention in applied prediction problems. Would love to hear your thoughts about this

maybe:The confidence approach can also be framed as an instance of the classifier/ rejector approach, where the rejector is equal to the classifier minus the threshold.

1. But, since the two models are not jointly learned – little can be said about the theoretical properties of such system. Unfortunately, even for a joint learning of the classifier and rejector, no theoretical guarantees are available for multiclass classification.