Crowd-Machine Collaboration for Item Screening

Evgeny Krivosheev
Bahareh Harandizadeh
University of Trento, Italy
first.last@unitn.it

Fabio Casati
University of Trento, Italy and
Tomsk Polytechnic University, Russia
first.last@unitn.it

Boualem Benatallah
UNSW, Sydney, Australia
boualem@cse.unsw.edu.au

ABSTRACT

In this paper we describe how crowd and machine classifier can be efficiently combined to screen items that satisfy a set of predicates. We show that this is a recurring problem in many domains, present machine-human (hybrid) algorithms that screen items efficiently and estimate the gain over human-only or machine-only screening in terms of performance and cost.

KEYWORDS

crowdsourcing, machine learning, hybrid systems, classification

1 BACKGROUND AND MOTIVATION

A frequently occurring classification problem consists in identifying items that pass a set of screening tests (filters). This is not only common in medical diagnosis but in many other fields as well, from database querying - where we filter tuples based on predicates [7], to hotel search - where we filter places based on features of interest [5], to systematic literature reviews (SLR) - where we screen candidate papers based on a set of criteria to assess whether they are in scope for the review [2]. The goal of this paper is to understand how, given a set of trained classifiers whose accuracy may or may not be known for the problem at hand (for a specific query predicate, hotel feature, or paper topic), we can combine machine learning (ML) and human (H) classifiers to screen items efficiently in terms of cost of querying the crowd, while ensuring an accuracy that is acceptable for the given problem. To make the paper easier to read and the problem concrete, we take the example of SLR mentioned above, which is rather challenging in that each SLR is different and each filtering predicate (called *exclusion criterion* in that context) could be unique to each SLR (e.g., "exclude papers that do not study adults 85+ years old"). Abundant prior art discusses crowd-based filtering (e.g., [2–7]), while research on hybrid classification is still in its infancy. Recent papers address the problem of combining machine and crowd intelligence in crowd-powered bots[8] as well as in crowdsourced classification[2]. The problem we address here differs from prior art in that i) we use the information provided by each kind of classifier (machine and human) to improve the effectiveness of the other kind so that they can be stronger together and ii) we consider a probabilistic model that works on a per filter and per item basis to minimize the overall number of crowd votes.

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186946

2 PROBLEM STATEMENT AND MODEL

We assume in input a set of items $i \in I$ to classify (in our example, these are papers to screen), a set of filters $f \in F$ (paper exclusion criteria), a set of ML or H classifiers $c \in C$, and a loss function $L = k * FE + FI$, modeled as a linear combination of false exclusions FE and false inclusions FI, which may carry a different relative weight k (e.g., most authors consider excluding relevant papers a more costly error than including an irrelevant one). Each classifier $c = \{cost, a(f), \rho(f, C)\} \in C$ is associated with the cost of asking one vote on an (item, filter) pair, with a filter-specific estimated accuracy (a 2x2 confusion matrix capturing probability of correct decisions for positive and negative labels), and with its correlation ρ with other classifiers. We specify filter-specific accuracy as we have seen that accuracy can vary greatly based on the filter (exclusion criteria) to be evaluated, for both machines (as studied in the experiments described later) and humans (as we reported in [3]). We do not discuss here how to obtain ML classifiers as this is the subject of ample literature: we merely assume they are given, and that we may or may not have information on their accuracy and correlation when applied to specific problem (our set of candidate papers and exclusion criteria). Consequently, we model accuracy as a beta distribution, where we incorporate prior knowledge if available, else we assign an initial uniform $Beta(1, 1)$ distribution for both positively and negatively labeled items. To simplify the presentation we assume to have three kinds of classifiers: machines (with zero cost per vote and $Beta(1, 1)$ accuracy), crowd (with cost 1 and also $Beta(1, 1)$ accuracy), and experts, with expert cost ec to which for simplicity we assign perfect accuracy. Consistently with crowdsourcing literature, we also assume that crowd and experts' opinions are independent, while in general we cannot make this assumption for ML classifiers. Our goal is, given quality parameters such as the loss function, to identify a strategy that can efficiently (in terms of cost) query the classifiers available and aggregate results while achieving the quality goals.

3 STRATEGIES AND EXPERIMENTS

We base the hybrid machine-crowd classification strategy on modifying the *shortest run* (SR) algorithm, developed for crowd-only classification [3]. SR proceeds by obtaining a test dataset T from "expensive" experts (usually 10-20 items) used to filter out low accuracy crowd workers, and by performing a *baseline run* with the - cheaper - crowd classifying a set of B items (usually 50-100) to estimate crowd accuracy and filter selectivity. Based on this estimate, and on the (initially empty) set of votes for the items to be classified, it then decides which filter to apply first to which item, to maximize the probability of screening the item out with few votes. It also estimates the expected cost for crowd classification, leaving items to experts when convenient [3]. We extend SR because i) it

was designed for multi-filter screening and has shown to perform better than baseline algorithms for crowd classification, ii) it has a per paper and per item probabilistic model that can leverage prior knowledge on items and filters, and ML classifiers can provide such knowledge, and iii) the algorithm can work with different sizes for test T and baselines B. This is important as test items can help us filter out ML classifiers with an expected accuracy lower than a threshold $\overline{a}$ (to be tuned as discussed later), and the more extensive set of crowd-classified items from the baseline can be used to a) assess independence among ML classifiers (which is necessary if we want to pool votes from ML classifiers with simple algorithms such as majority voting), and b) build an ensemble model out of the ML classifiers where the output of each ML classifier is a feature [1]. Therefore, the classification strategy proceeds as follows:

(1) Obtain gold dataset T from expert and use it to screen ML classifiers and to use as tests questions for crowd workers. If we start from a Beta(1,1) uniform prior for a given filter, we know the accuracy distribution goes to a $Beta(1+correct\_answers, 1+failed\_answers)$ which has a known pdf and mean.
(2) Perform a baseline run on B items (on all filters), both to estimate crowd accuracy on each filters and to get data for the next step.
(3) Compute correlation among ML classifiers and remove classifiers with correlation higher than a threshold c that we tune empirically, so that we can meaningfully use weighted majority voting to combine the opinions of different ML classifiers.
(4) Compute the probability that a filter applies to an item by combining the vote of the ensemble ML classifier (which now has a known accuracy). Treat this value as a prior probability for the (filter, item) pair and continue with SR until items are classified by the crowd or until SR decides they are to be left to experts.

Many variations are possible over this basic scheme, including changing the size of test data T and baseline B as well as building a model (e.g., logistic regression) to combine classifiers votes as opposed to using majority voting, leveraging the baseline run.

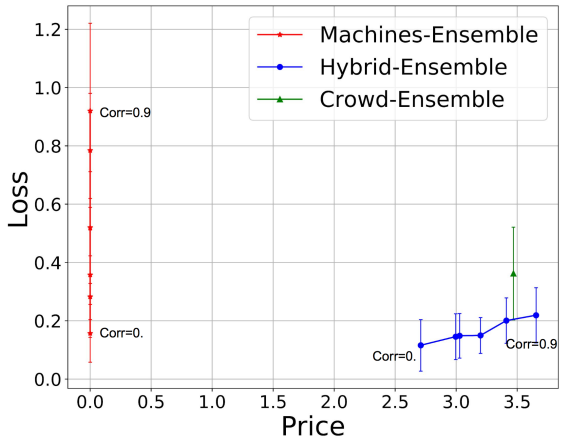

Figure 1: Expected loss vs price for different algorithms. Correlation values between machines = [0., 0.2, 0.3, 0.5, 0.7, 0.9]

Experiments. To assess the approach, we ran experiments on Mechanical Turk (described in [3]) as well as leverage existing crowd datasets [6]. In both cases the experiments are related to SLRs with multiple filters and include over 20000 crowd votes on over 4000 papers. We refer to the cited papers for details on experiment design. We used these datasets to get realistic data on crowd worker accuracies, on variation of such accuracies by filter and on filter power. czWe then built classifiers for each filter using a variety of techniques (from KNN to random forest, variations of naive Bayes, and others[1]) and different sizes of training data to get realistic information on classifier accuracies and correlations. We obtained classifier accuracies in the 0.5-0.95 range and correlations in the 0.2-0.9 range, and crowd accuracy in the 0.55-0.8 range. We then used this data to simulate a variety of scenarios. In Figure 1 we compare the results of applying machine only, crowd (with SR), and hybrid strategy, to simulations of classifications for 1000 papers and 4 filters (averages over 50 iterations). We simulate 10 ML classifiers with accuracy randomly selected from a 0.5-0.95 range and on which the algorithm assumes no prior knowledge. We screen them with T=20 tests and keep the ones with 0.95 probability of having an accuracy greater than 0.5. The threshold for false exclusion error is set at 0.01 as in [3] and the weight K in the loss function is set to 5. We then plot the average loss and price paid per item as the correlation among classifiers vary from 0 to 0.9 (with price growing with the correlation). As we can see, for a similar loss level, hybrid algorithms significantly outperforms the crowd in terms of price, with savings from 7.4% to 53.7% depending on correlation. Notice that we disregard here the cost of obtaining the classifier, which, if they are built from scratch for this specific SLR, needs to be factored in when estimating price and assessing the best strategy. If classifiers accuracies worsen (e.g., lie in the 0.4-0.8 or 0.3-0.7 range), the savings also decrease approximately by a factor of 2 and 4 respectively. For near-zero correlation (very hard to achieve in practice) the ML-only strategy becomes appealing, and then deteriorates. The reader can see on the GitHub repository in footnote how results vary as we change parameters and thresholds.

Acknowledgements. This project has received funding from the EU Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690962.

REFERENCES

[1] Saso Džeroski and Bernard Ženko. 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning* 54, 3 (01 Mar 2004), 255–273.
[2] Byron C Wallace et al. 2017. Identifying reports of randomized controlled trials (RCTs) via a hybrid machine learning and crowdsourcing approach. *J Am Med Inform Assoc* (2017).
[3] Evgeny Krivosheev, Boualem Benatallah, and Fabio Casati. 2018. Crowd-based Multi-predicate Screening of Papers in Literature Reviews. In *Proceedings of WWW2018*. International World Wide Web Conferences Steering Committee.
[4] Evgeny Krivosheev, Valentina Caforio, Boualem Benatallah, and Fabio Casati. 2017. Crowdsourcing Paper Screening in Systematic Literature Reviews. In *Procs of Hcomp2017*. AAAI.
[5] Doren Lan, Katherine Reed, Austin Shin, and Beth Trushkowsky. 2017. Dynamic Filter: Adaptive Query Processing with the Crowd. In *Procs of Hcomp2017*. AAAI.
[6] Michael L. Mortensen, Gaelen P. Adam, Thomas A. Trikalinos, Tim Kraska, and Byron C. Wallace. 2016. An exploration of crowdsourcing citation screening for systematic reviews. *Research Synthesis Methods* (2016). RSM-02-2016-0006.R4.
[7] Aditya Parameswaran, Stephen Boyd, Hector Garcia-Molina, Ashish Gupta, Neoklis Polyzotis, and Jennifer Widom. 2014. Optimal crowd-powered rating and filtering algorithms. In *Proceedings of VLDB*. VLDB Endowment.
[8] Denis Savenkov and Eugene Agichtein. 2016. CRQA: Crowd-powered Real-time Automatic Question Answering System. In *Procs of Hcomp2016*. AAAI.

[1] see jointresearch.net for details