Anomaly Detection in Crowdsourcing: Why Midpoints in Interval-Valued Approach

Alejandra De La Peña, Damian L. Gallegos Espinoza, and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso
El Paso, Texas 79968, USA
adelapena5@miners.utep.edu, dlgallegose@miners.utep.edu
vladik@utep.edu

1. What is crowdsourcing: a brief reminder

- In many practical situations, we need to perform a large number of reasonably simple tasks, tasks that do not require high qualifications.
- For example, deep learning requires that a large number of labeled examples by available.
- In many cases, we do not have that many labeled examples.
- So we need someone to label a large number of photos, or a large number of speech recordings.
- One way to perform these tasks is *crowdsourcing*, when people all over the world are paid to solve the corresponding tasks.
- Example: label pictures that will be used for training a machine learning algorithm.

2. Need to detect anomalies

- Most crowd-workers work conscientiously; however:
  - since the payment is proportional to the number of answers,
  - there are also many cases when crowd-workers do a sloppy job,
  - not spending enough time on analyzing the corresponding picture and
  - therefore producing answers that are often wrong.
- Such wrong answers prevent machine learning algorithms from getting high quality results.
- It is therefore important to be able to detect such anomalous crowd-workers and dismiss their answers.

3. Need to detect anomalies (cont-d)

- A natural way to do it is to include examples with known labels into the list of tasks.
- Then, we can gauge the quality of a crowd-worker by the number of wrong answers that he/she has on these examples.
- If this number is unusually high, then all the answers provided by this crowd-worker should be dismissed.

4. Need to take into account degrees of confidence

- Crowd-workers are often not 100% confident in their answers.
- To help machine learning, it is therefore desirable to collect:
  - not only the answers,
  - but also the degrees indicating how confident is the crowd-worker in each answer.
- This way, the neural network will be able to weigh these answers with different weights:
  - if its answer is different from the confident answer of a crowd-worker, then the algorithm should continue training, but
  - if the difference is only with not very confident crowd-workers, then maybe there is no need to adjust.

5. Need to take into account degrees of confidence (cont-d)

- Because of this, some crowdsourcing algorithms require the crowd-worker to submit:
  - not only the answer,
  - but also his/her degree of confidence in this answer – as expressed by a number on some scale $[\underline{X}, \overline{X}]$,
  - e.g., from 0 to 10, or from 0 to 1.
- Usually, larger numbers correspond to larger degrees of confidence.
- Usually, linear transformations are used to transform between different scales.
- For example, the value 7 on a scale from 0 to 10 is transformed into 7/10 on the scale from 0 to 1.
- Similarly, the value 0 on the scale $[-1, 1]$ is transformed into the value 0.5 on the scale $[0, 1]$.

6. Need to take into account degrees of confidence (cont-d)

- These degrees of confidence are used to detect anomalies.
- If the answer is wrong but the crowd-worker is not very confident about it, this may be an honest mistake.
- However, if there are many wrong answers with high degrees of confidence, this indicates an anomaly.
- Sometimes, these degrees also affect the amount of payment:
  - the higher degree of confidence, the higher the pay –
  - since one way to gain more confidence is to spend more time analyzing the corresponding picture or recording.

7. Interval-valued degrees of confidence

- Crowd-workers are usually unable to describe their degree of confidence by a single number.
- In general, people cannot meaningfully distinguish, e.g., between degrees of confidence 0.70 and 0.71 on a scale from 0 to 1.
- So, it makes sense to allow the crowd-workers to mark their confidence by selecting an interval $[\underline{x}, \overline{x}]$ of possible degrees.
- For example, an interval $[0.7, 0.8]$.

8. How to detect anomalies based on interval-valued degrees: formulation of the problem

- A natural idea is to utilize formulas that have been successful in detecting anomalies based on numerical degrees.
- To apply these formulas, we need to select a single value x from the corresponding interval $[\underline{x}, \overline{x}]$.
- In other words, we need an algorithm $x = f(\underline{x}, \overline{x})$ that generates a number based on the bounds of the worker-generated interval.
- Which algorithm $f(\underline{x}, \overline{x})$ should we select?
- We can take arithmetic average, we can take geometric average $\sqrt{\underline{x} \cdot \overline{x}}$, we can have many other choices.

9. How to detect anomalies based on interval-valued degrees: formulation of the problem (cont-d)

- An empirical analysis has shown that the more accurate anomaly detection happens when we use arithmetic average

$$\frac{\underline{x} + \overline{x}}{2}.$$

- How can we explain this empirical result?
- In this talk, we use natural invariances to explain this empirical result.

10. Invariance

- We can have different scales.
- So it is reasonable to require that the desired algorithm $x(\underline{x}, \overline{x})$ should not change if we apply some linear transformation to a different scale.
- Thus, we arrive at the following definition.
- We say that a function $f(\underline{x}, \overline{x})$ is *scale-invariant* if for every linear transformation $x \mapsto a + b \cdot x$ with $b > 0$, and for all possible $\underline{x} < \overline{x}$:
  - once we have $x = f(\underline{x}, \overline{x})$,
  - then we should also have $y = f(\underline{y}, \overline{y})$, where $y = a + b \cdot x$, $\underline{y} = a + b \cdot \underline{x}$, and $\overline{y} = a + b \cdot \overline{x}$.

11. Additional requirement related to negation

- Sometimes, there are only two possible choices A and B.
- If we use the scale from 0 to 1, then one way to interpret the degree of confidence x is as the probability that the correct choice is A.
- This same situation can be interpreted as the probability $1 - x$ that the correct choice is B.
- What if, instead of the exact probability x, we have an interval $[\underline{x}, \overline{x}]$ of possible values of A-probability?
- Then the corresponding values of B-probability $1 - x$ form an interval $[1 - \overline{x}, 1 - \underline{x}]$.
- We can apply the desired function to the original interval $[\underline{x}, \overline{x}]$ and thus get some probability x.
- Alternatively, we can apply the same function to the negation-related interval $[1 - \overline{x}, 1 - \underline{x}]$ and get some probability y.
- In this case, for the probability of A, we get the value $1 - y$.

12. Additional requirement related to negation (cont-d)

- Since these are two ways to describe the same situation, it is reasonable to require that we should get the same probability.
- So, we should have $x = 1 - y$.
- Thus, we arrive at the following definition.
- We say that a function $f(\underline{x}, \overline{x})$ is *negation-invariant* if for all possible values $0 \le \underline{x} < \overline{x} \le 1$:
  - once we have $x = f(\underline{x}, \overline{x})$,
  - then we should also have $y = f(\underline{y}, \overline{y})$, where $y = 1 - x$, $\underline{y} = 1 - \overline{x}$, and $\overline{y} = 1 - \underline{x}$.

13. Proposition

The only scale-invariant and negation-invariant function $f(\underline{x}, \overline{x})$ *is arithmetic average* $f(\underline{x}, \overline{x}) = \dfrac{\underline{x} + \overline{x}}{2}$.

14. Proof

- It is easy to check that arithmetic average is scale-invariant and negation-invariant.
- Let us prove that, vice versa, every scale-invariant and negation-invariant function $f(\underline{x}, \overline{x})$ is arithmetic average.
- Indeed, let us denote $f(0, 1)$ by α.
- Here, $\underline{x} = 0$, $\overline{x} = 1$, and $x = \alpha$.
- Then, for every two numbers $x_1 < x_2$, we can take $a = x_1$ and $b = x_2 - x_1$.
- In this case, $\underline{y} = a + b \cdot \underline{x} = x_1$,

$$\overline{y} = a + b \cdot \overline{x} = x_1 + (x_2 - x_1) = x_2, \text{ and}$$

$$y = a + b \cdot x = x_1 + \alpha \cdot (x_2 - x_1) = \alpha \cdot x_2 + (1 - \alpha) \cdot x_1.$$

- Thus, due to scale-invariance, we conclude that

$$f(x_1, x_2) = \alpha \cdot x_2 + (1 - \alpha) \cdot x_1.$$

15. Proof (cont-d)

- To find α, let us now use negation invariance.
- According to this property, we should have
$$f(1-x_2, 1-x_1) = 1 - f(x_1, x_2).$$
- Substituting the expression for f into this formula, we conclude that
$$\alpha \cdot (1-x_1) + (1-\alpha) \cdot (1-x_2) = 1 - \alpha \cdot x_2 - (1-\alpha) \cdot x_1.$$
- If we open parentheses, we conclude that
$$1 - \alpha \cdot x_1 - (1-\alpha) \cdot x_2 = 1 - \alpha \cdot x_2 - (1-\alpha) \cdot x_1.$$
- The two linear functions on both sides of this formula should be equal to all $x_1 < x_2$.
- Thus, the coefficients at x_1 must coincide, so $\alpha = 1 - \alpha$.
- Thus, $\alpha = 1/2$ – and therefore, the above formula becomes arithmetic average.
- The proposition is proven.

16. Conclusion

- We have explained why arithmetic average works well:
- It is the only function that satisfies natural invariance requirements.

17. Acknowledgments

- This work was supported in part by the National Science Foundation grants:
  - 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and
  - HRD-1834620 and HRD-2034030 (CAHSI Includes).
- It was also supported by the AT&T Fellowship in Information Technology.
- It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.