7-2020

Adversarial Teaching Approach to Cybersecurity: A Mathematical Model Explains Why It Works Well

Christian Servin

Olga Kosheleva

Vladik Kreinovich

Comments:

Technical Report: UTEP-CS-20-78

Adversarial Teaching Approach to Cybersecurity: A Mathematical Model Explains Why It Works Well

Christian Servin
Computer Science and
Information Technology Systems Department
El Paso Community College (EPCC)
919 Hunter Dr., El Paso, TX 79915-1908, USA
cservin1@epcc.edu

Olga Kosheleva
Department of Teacher Education
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
olgak@utep.edu

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
vladik@utep.edu

***Abstract*—Teaching cybersecurity means teaching all possible ways how software can be attacked – and how to fight such attacks. From the usual pedagogical viewpoint, a natural idea seems to be to teach all these ways one by one. Surprisingly, a completely different approach works even better: when the class is divided into sparring mini-teams that try their best to attack each other and defend from each other. In spite of the lack of thoroughness, this approach generates good specialists – but why? In this paper, by analyzing a simple mathematical model of this situation, we explain why this approach work – and, moreover, we show that it is optimal in some reasonable sense.**

***Keywords*—Teaching cybersecurity, adversarial thinking, optimal teaching.**

I. Formulation of the Problem

Cybersecurity is important. In the modern world, everything relies on computers – even more so with the current COVID'19 pandemic. Computers run our communications, computers control our utilities, computers largely control our planes, cars, etc. For our civilization to continue to function, it is important to protect all these computer systems from malicious attacks.

Teaching cybersecurity is important. Whatever automatic tools we place in to prevent cyber-attacks, smart adversaries learn to overcome. The only way to maintain cybersecurity is to train a large corpus of specialists who would protect us from all the newly appearing threats.

Traditional way to teaching – in particular, to teaching cybersecurity. The usual way of teaching any material is to present, to the students, the needed information and skills. With respect to cybersecurity, this means explaining, to the students, the main types of cyber-attacks and the main ways to defend against these attacks. After that, we can let the students show their creativity, but usually, teaching the basics is a must.

Adversarial teaching: a successful alternative approach. Interesting, lately, a different approach has been very popular and very successful, in which, instead of teaching students the usual way, the instructor divides the class into one or more pairs of sparring mini-teams. In each pair, the teams interchangingly try to attack each other and to defend their team from a partner's attacks.

This works, but why? The above strategy works, which is somewhat surprising. In the absence of a thorough coverage of all possible topics, one would expect gaps in the ability of students who have been taught this way – but there are usually no such gaps. So, the first question is: why this approach works?

A natural second question: is this approach close to optimal or we can drastically further improve it – and if yes, how?

What we do in this paper. In this paper, we answer both questions: we explain why the adversarial teaching approach works, and we show that this approach is – in some reasonable sense – optimal.

II. Analysis of the Problem

Similar approach works in design. For teaching, this approach may be somewhat new, but a similar approach works in military engineering. For example, according to [5], new fighter planes are designed as follows (by using a program that simulates dogfights between different planes):

The first stage is natural: we consider several possible designs, and for each of them, we simulate how this design will perform in a possible confrontation with the fighter planes used by the existing adversaries. We continue doing this until we find a design that can beat all the possible opponents.

At first glance, this may seem to be sufficient, but, on second thought, it is not: it is not enough for a future plane to be better that what the opponent has now, we need to have a design that will be better than what the opponent will have in the future.

To design such a plane, we perform the second stage of the design process: namely, we design a plane that will be better than not only the current planes, but also better than our first-stage design.

Then, we design a plane that will be better than the second-stage design, etc. At the end, we get an almost perfect future plane – and this is what is then implemented and tested.

What can we conclude from this fact. The fact that a similar idea works successfully in such completely different application areas as teaching cybersecurity and designing

fighter planes makes us confident that these successes are not due to any specific features of these areas, they are due to the general structure of this approach. Let us therefore describe a simple mathematical model that would capture this structure.

Comment. We are not specialists in plane design, but, as educators, we are clearly more familiar with educational applications. So, while the model will be potentially general, we will illustrate it on the example of teaching – namely, on the example of teaching cybersecurity.

Towards a model. We want the students to be able to handle all possible attack situations. Of course, different situations are all somewhat different, but ideally, what we want is to make sure that whatever new situation surfaces, the students should have some experience successfully fighting a similar attack in the past, experience that would help the student fight the new attack as well.

In mathematics, a natural way to describe similarity is by assuming that there is a some metric $d(a,b)$ on the set S of possible situations, a metric that describes to what extent situations a and b are different from each other – or similar to each other. The smaller the distance $d(a,b)$, the more similar are situations a and b.

In these terms, "similar" means that the distance $d(a,b)$ is smaller than or equal to some small threshold value $\varepsilon > 0$.

Therefore, we arrive at the following model.

III. Resulting Model: What We Have and What We Want

Resulting model. The above requirement can be formulated as follows. We have a set S of possible situations. On this set, we have a metric $d(a,b)$.

We want the student to experience situations $s_1, \ldots, s_n$ such that every situation s from the set S is ε-close to one of such situations.

Comments.

- In mathematics, such a set is known as an *ε-net*; see, e.g., [3], [4].
- The exact value of the threshold is determined by our resources: the smaller ε, the better – but a drastic decrease in ε would mean a drastic increase in situations experienced during teaching, and the teaching time is limited.

How do we compare quality of different teaching schemes. In view of the previous comment, once we fix $\varepsilon > 0$, a natural measure of quality is the number of experiences situations n: the smaller n, the faster we can train.

Alternatively, we can fix n – and thus, the training time – and try to find the situations $s_1, \ldots, s_n$ that lead to the smallest possible ε.

Comment. For each metric space, the smallest possible number of elements in an ε-net is called *ε-entropy*; to be more precise, usually the logarithm of this smallest number is called the ε-entropy; see, e.g., [3], [4].

The corresponding optimization problem is known to be NP-hard. It is known that problem of finding the smallest ε-net is, in general, NP-hard; see, e.g., [1]. This means, crudely speaking, that unless P = NP (which most computer scientists believe to be false), no feasible algorithm is possible that would always find the optimal ε-net.

IV. Adversarial Teaching Reformulated in Terms of the Model: Formulation and Analysis

Let us reformulate adversarial teaching in these terms. The first team starts with some attack situation s_1. Then, the sparring team learns how to defend against this attack. So, next time, the attacking team will try to find a new way of attacking that has the most chances of success – i.e., the situation s_2 which is as far away from the original situation s_1 as possible:

$$d(s_2, s_1) = \max_{s \in S} d(s, s_1). \tag{1}$$

Once the sparring team learns how to deal with the situation s_2 as well, the next attacking situation s_3 will be as far away from both s_1 and s_2 as possible, i.e., for which the distance

$$d(s, \{s_1, s_2\}) \stackrel{\text{def}}{=} \min(d(s, s_1), d(s, s_2)) \tag{2}$$

is the smallest possible:

$$\min(d(s_3, s_1), d(s_3, s_2)) = \max_{s \in S} \left(\min(d(s, s_1), d(s, s_2))\right). \tag{3}$$

In general, once we have experiences the situations $s_1, \ldots, s_k$, we select the next situation s_{k+1} for which

$$\min(d(s_k, s_1), \ldots, d(s_k, s_{k-1})) = \max_{s \in S} \left(\min(d(s, s_1), \ldots, d(s, s_{k-1}))\right). \tag{4}$$

We continue until this way, we can find a situation which is different from all the previous ones – i.e., for which $d(s_k, s_i) > \varepsilon$ for all $i < k$.

When this is no longer possible, i.e., when we have

$$\max_{s \in S} \left(\min(d(s, s_1), \ldots, d(s, s_n))\right) \le \varepsilon, \tag{5}$$

we stop.

This strategy works: a proof. There are only finitely many possible situation – e.g., since each situation has to be described in a reasonable time and thus, contain a reasonable number of characters N to describe, and for each N and for each set of possible symbols, we have a finite number of strings of this (or smaller) length.

At each iteration, we generate a situation which different from all the previous once. Thus, eventually, the above process will stop.

Let us show that the resulting set of situations $s_1, \ldots, s_n$ indeed satisfies the desired property – that every situation $s \in S$ is ε-close to one of the situations s_i.

Indeed, the formula (5) means that for every situation $s \in S$, we have

$$\min(d(s, s_1), \ldots, d(s, s_n)) \le \varepsilon, \tag{6}$$

which, in its turn, means that for every situation $s \in S$, there exists a situation i for which $d(s, s_i) \le \varepsilon$.

This strategy is asymptotically optimal: formulation. Let n be the number of situations that the students have experienced by following this strategy. As we have mentioned earlier, because the strategy is feasible and the problem is NP-hard, we cannot expect that for this number n, the threshold ε is optimal. It is thus possible that, in principle, with the same number n, we can reach a smaller value ε'.

What we *can* prove, however, is that this decrease cannot be too drastic: namely, we will prove that even for one fewer $(n-1)$ situation, the corresponding optimal value ε' is at best twice smaller, i.e., that $\varepsilon' \ge \varepsilon/2$.

This strategy is asymptotically optimal: a proof. Let us prove this optimality result by contradiction.

Indeed, by our construction, we have

$$d(s_i, s_j) \ge \varepsilon \tag{7}$$

for all $i \ne j$. Suppose that we have a ε'-net $s'_1, \ldots, s'_{n-1}$. By definition a ε'-net, each element s_i is ε'-close to some element $s'_{e(i)}$.

For $i \ne j$, we cannot have $e(i) = e(j)$: otherwise, we will have

$$d(s_i, s_j) \le d(s_i, s_{e_i}) + d(s_j, s_{e_i}) \le 2\varepsilon' < \varepsilon, \tag{8}$$

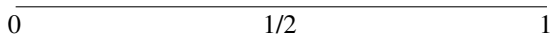
which contradicts (7). Thus, to each of the n elements s_i, we assign a different element s'_j – but this is impossible, since we assumed that we only have $n-1$ elements e'_j.

The optimality is thus proven.

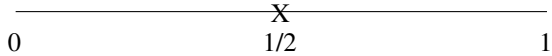
V. Graphical illustration

To make it easier to understand, let us give two simple geometric illustrations of the above idea. These examples are similar to examples provided in [2] for a different application of a similar idea – to selecting benchmarks for testing different numerical algorithms.

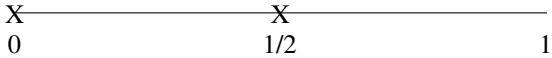
1D example. Let us start with the simplest example of a metric space S – namely, the interval $[0, 1]$:



It is reasonable to select the midpoint 1/2 as s_1:

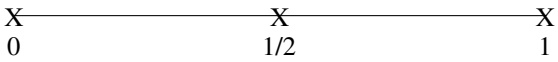


There are two points that are the farthest from s_1: the left endpoint 0 and the right endpoint 1. Without losing generality, let us select $s_2 = 0$:

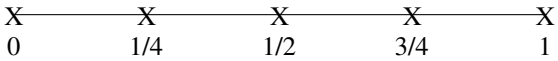


Now, $s_3 = 1$ is the point with the largest value of

$$d(s, \{s_1, s_2\}) = \min(d(s, s_1), d(s, s_2)) :$$



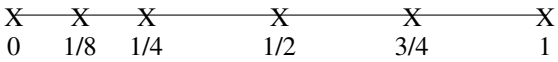
At this stage, the midpoints between 0 and 1/2 and between 1/2 and 1 are the farthest from the set $\{s_1, s_2, s_3\} = \{0, 1/2, 1\}$, so, after two stages, we add them both:



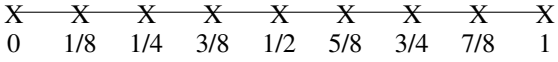
Now, the largest possible value of

$$d(s, \{s_1, s_2, s_3, s_4, s_5\}) = d(s, \{0, 1/4, 1/2, 3/4, 1\})$$

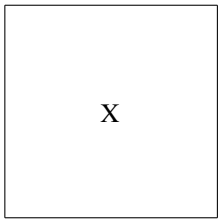
is 1/8. So, at the next stage, we add one of the points in between the existing ones, e.g., the first one (1/8):



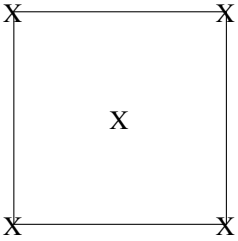
After three more stages, we add all midpoints, so we arrive at the following configuration:



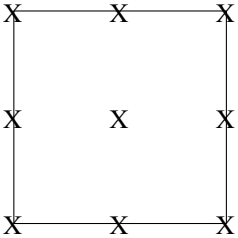
2D example: square. For a unit square, we get a similar situation. First, let us pick the midpoint as s_1:



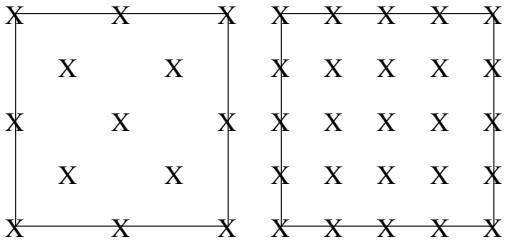
Then, the next four selections s_i are the vertices:



After this, the next four selected points s_i are he midpoints of the four edges:



Here, we have, in effect, four sub-squares. On the next stage, the same procedure is repeated for each sub-square, etc.



ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

REFERENCES

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, and Mc-Graw Hill Co., N.Y., 2009.

[2] V. Kreinovich and S. A. Starks, "Why benchmarking is an (asymptotically) optimal approach to numerical methods: a geombinatoric proof", *Geombinatorics*, 2004, Vol. 13, No. 3, pp. 131–138.

[3] G. G. Lorentz, *Approximation of Functions*, Halt, Reinhart, and Winston, New York, 1966.

[4] G. G. Lorentz, "The 13-th problem of Hilbert", in: F. E. Browder (ed.), *Mathematical Developments Arising from Hilbert's Problems*, American Math. Society, Providence, Rhode Island, 1976, Part 2, pp. 419–430.

[5] N. Moiseev, *Elements of the Theory of Optimal Systems*, Moscow, Nauka, 1975 (in Russian)