MathAData: Teaching Mathematics through Experimentations with AI Challenges

Stéphane Mallat
Chair of Data Science
Collège de France

1 Introduction

Numerous studies show that the back-and-forth between phases of experimentation, abstraction, and mathematical processing plays a fundamental role in improving the understanding of mathematical tools [3]. This finding is rarely questioned and appears in one of the first proposals of the Torrossian-Vilani report [1] on the teaching of mathematics. This approach is successfully implemented in primary education in some countries, notably inspired by the "Singapore method." It links the mathematical tools students learn (numbers, addition, multiplication) with their intuition in practical situations where these mathematics apply. This significantly deepens the understanding of abstract mathematical concepts.

In primary school, experimentation can be done through object manipulation or games. While this pedagogical approach remains just as valid in high school, experimentation becomes more complex. It must be related to the higher-level mathematics in the curriculum. The problems posed must also be motivating for teenagers and help them envision a future connected to mathematics. This is crucial for reducing the decline in interest in mathematics between the sophomore and senior years of high school, which is observed today in France. It is about showing that mathematics is not disconnected from the world and plays an important role in addressing major societal issues, from health to climate change.

The back-and-forth between experimentation, abstraction, and mathematics is also at the heart of mathematical research [2]. Questions posed by the sciences, notably physics, have always been a fundamental driver for the development of new mathematics, as seen in the works of Newton, Gauss, Fourier, Poincaré... The beauty of mathematics then appears in the simplicity and power of abstract concepts that solve seemingly very different practical problems. However, it is not easy to teach high school mathematics with an experimental approach in physics, chemistry, or biology. This requires students to understand these other sciences well enough, and the time spent on scientific experimentation reduces the time dedicated to mathematics. This article explains why data challenges can circumvent these difficulties while posing questions on a variety of topics. As such, they open up new opportunities for teaching mathematics in high school.

The challenge is not only on the students' side. In France, many teachers have been trained with a relatively formal approach to mathematics. Indeed, mathematics can be introduced through an axiomatic approach, rooted in set theory. However, this approach has proven to be a very difficult path for teaching, making mathematics inaccessible to many students. Even though the national education system has long since moved away from this approach, this culture and the prestige of formalism remain present in French university teaching. The connection of mathematics with the world is then taught through a few "applications" at the end of the course, if there is time left.

Teaching mathematics through a back-and-forth with questions arising from experiments, with phases of modeling and mathematical processing, is a profound change. Working on open problems, which admit multiple solutions, is also important for students to express their creativity. This prevents them from being stuck or obsessed with finding "the answer," engaging instead in a less stressful trial-and-error process.

Here, we see that numerous obstacles accumulate to develop such a teaching of mathematics, which is nevertheless crucial. The constraints can be summarized as follows:

1. Open problems, with significant stakes and adapted to students' interests, but which can be quickly explained in mathematical terms.

2. Problems that involve a large part of the mathematics in the curriculum, with a modeling phase, and that open up a perspective on the importance of more sophisticated mathematics for solving important questions.

3. Experiments that fit into a balanced back-and-forth with mathematical abstraction and processing.

4. To scale nationally, it must adapt to the diversity of students' levels and train teachers.

The objective of this article is to show that data challenges open up new perspectives for tackling all aspects of this problem head-on, with digital experiments.

MathAData and National Education Co-development MathAData brings together a team from the Collège de France and the École Normale Supérieure, in partnership with the Institut Louis Bachelier, which organizes AI data challenges on the web platform *challengedata.ens.fr.* The ambition of this team is to propose a pragmatic approach to developing the teaching of mathematics through experimentation, with data challenges, in high school and university. To adapt such an approach to high school, MathAData is conducting significant pedagogical and didactic work with teachers in classrooms. This work is carried out in co-development with the math labs of the French national education system in Lille [5].

Artificial intelligence is already present in high school, where more and more students use ChatGPT to do their homework at home. Studying these challenges also helps to demystify artificial intelligence by showing its mathematical roots, as explained in section 2. Section 3 shows that solving challenges covers most of the mathematics taught in high school: statistics, probabilities, algebra, analysis, and geometry. This allows these mathematics to be approached through digital experiments. Section 4 addresses the difficulties of such teaching in classrooms and the proposed pedagogical solutions.

2 AI Data Challenges

The objective of this first section is to show that data challenges are open problems, with stakes adapted to students' interests, and can be quickly explained in mathematical terms.

Data Challenges An AI challenge consists of developing an algorithm that can learn to answer a posed question based on data, using provided training examples. The mathematical formalization is identical for all challenges. The description below is for mathematics teachers, not their students.

Let d denote the data, for example, an image, specified by a list of numbers. Let r denote the response value, for example, 0 if it is a cat image and 1 if it is a dog image. From the data d, the goal is to compute an estimate $\hat{r}$ of the response r, with the least possible error. Classification, regression, and generation problems correspond to different types of responses r. In classification, we must identify the category of each data d, for example, the class of cat images versus dog images, and r is the class index coded by an integer. Regression estimates a real number r, such as the height of a person appearing in an image. A generation problem estimates a new data r that can be high-dimensional, such as a new image, or a text r generated from a text d that poses a question. Although classification, regression, and generation challenges appear very different, they are solved in the same mathematical framework introduced in the next section.

In all challenges, the data d is a list of d real numbers $d = (d_1, d_2, ...)$. If d is a grayscale image, each d_i specifies the light intensity of a point (pixel) in the image, coded by an integer between 0 (black pixel) and 255 (white pixel). The dimension of d is large, typically from hundreds to several million variables for an image.

Figure 1 shows two classification challenges. For the challenge called MNIST, the data d is a small grayscale image of a handwritten digit, with 28×28 pixels. The challenge is to identify the value $r \in \{0, ..., 9\}$ of the digit appearing in the image. For the challenge called CIFAR, the data is a small color image with 32×32 pixels, to be classified into 10 categories: cats, horses, frogs, cars, boats... also indexed by an integer $r \in \{0, ..., 9\}$. This challenge is much more difficult than recognizing digits in MNIST, but it remains accessible to high school students using the same learning algorithms.

Figure 1: A classification challenge asks to estimate the class index r of a data d. Top: examples of images from the MNIST challenge. Different classes correspond to different handwritten digits. Bottom: examples of images from the CIFAR challenge. Image classes correspond to airplanes, cars, birds, cats...

Web Platform The platform *challengedata.ens.fr* offers more than 80 challenges of different types and difficulty levels for teaching in high school, university, and research. These can be image recognition problems, medical diagnostics, climate prediction, pollution calculation, text analysis... The data can then be images, time series, texts, sounds... The proposed challenges are used for university teaching from the bachelor's to the doctorate level, in all fields of science and humanities. The easiest ones are suitable for high school teaching.

For each challenge, the web platform provides training examples with the data d and responses r. For example, the class of the animal appearing in the image. We will see that the algorithm's parameters are optimized on these training examples. Like a teacher, the platform offers a test procedure on new data d different from the training examples, for which the response r is not provided. A participant submits to the platform the estimates $\hat{r}$ calculated by their algorithm on these test examples. The platform, which knows the correct answers, returns a score (a grade), related to the average error on the test examples. This score is displayed on a "leaderboard" to compare the effectiveness of different learning algorithms. This aspect gives a playful component to the search for solutions, which can be practiced in groups. A collaboration phase for the whole class often results in a better solution than the solutions found by each group.

A challenge is quickly explained. It suffices to introduce the data d and the response r. The link between the response r and the data d is not explained, as would be done in physics. It will be up to the students to discover it during their modeling, with learning algorithms that we will now explain.

3 The Mathematics of Artificial Intelligence

One might think that solving a data challenge is primarily a statistics problem. This is not at all the case. We will show that solving these challenges involves a large part of the mathematics in the high school curriculum. This brief presentation introduces both the principles of artificial intelligence and the mathematical fields involved. The pedagogical presentation for high school students simplifies and details this introduction, integrating it into the presentations of the mathematics chapters in the curriculum.

The goal is to find an algorithm that calculates the response r from the data d. For example, r can be the class index of the image d. To do this, we define an algorithm that calculates an estimate $\hat{r}$ of r, using a list of parameters $x = (x_1, x_2, ...)$ that need to be learned.

One of the important ideas is that the data is represented in a Euclidean space, where the distance is a measure of similarity between the data. As explained in section 3.1, this becomes a geometry problem. In high school, this space is a line, a plane, or three-dimensional space. Calculating the response associated with each data involves separating this Euclidean space into different zones. This separation is learned by setting the parameters x of the boundary between these zones. Section 3.2 explains that these parameters are optimized on training examples, so that the estimate $\hat{r}$ of the response r has as few errors as possible. This is similar to a student learning by practicing exercises with answers provided by their teacher, so they learn from their mistakes.

The goal is for the algorithm to make few errors on data different from the training data but of the same type. We say that the algorithm "generalizes". Section 3.3 shows that this is evaluated by calculating the error made by the algorithm on test data, different from the training data. This is the same approach as a teacher evaluating students' learning by giving an exam, with similar but different exercises from those studied in class.

3.1 Parametric Estimation and Euclidean Geometry

Features To calculate the response r associated with a data d, we associate with each data d a list of features $k = (k_1, k_2, ...)$. If we take 2 features, then k is considered as a point with coordinates (k_1, k_2) in a plane. More generally, k is a point in a Euclidean space whose dimension is equal to the number of features, which are the coordinates of k.

We will adjust the calculation of k so that data d with the same response r have features k that are close. Conversely, the features should be distant when the responses are different. Therefore, it is necessary to choose features that discriminate data corresponding to different responses. This is the main difficulty of each challenge.

As an illustration, we simplify the MNIST challenge by classifying only two handwritten numbers: images of 2 and 7. We can define a feature k_1 as the average intensity of the pixels $(d_1, d_2, ...)$ in the image. Images of 2 typically have more light pixels (high values) than images of 7. Therefore, they have a higher average value $k = k_1$, which allows them to be discriminated. However, a more discriminative feature can be found, for example, with an average localized in certain parts of the image. Many ideas can be tried, allowing students to find creative solutions.

The result can also be refined by taking two features instead of one. For example, k_1 can be the average in the upper half of the image and k_2 the average in the lower half of the image. Figure 2 shows all these features for images of 2 (in blue) and images of 7 (in orange) in a training dataset. We see that these two point clouds are quite well separated, but not perfectly. A third feature k_3 can be selected to better separate these images, in which case each feature is represented by a point in three-dimensional space. The more features we have, the better we can separate the data. The search for features is part of the creative work of students, allowing them to perform an initial modeling of the problem.

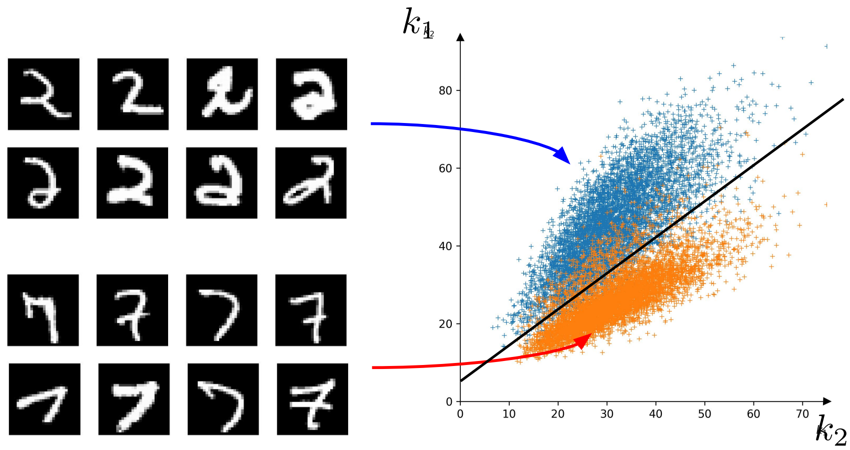

Figure 2: Each image is associated here with two features (k_1, k_2) which are the coordinates of a point in a plane. The features of images of 2 are blue points and those of images of 7 are orange points. A linear classifier separates the blue point cloud from the orange point cloud with a line. Errors correspond to points that are on the wrong side of the line.

Geometric Separation Building classifiers in the Euclidean feature space involves most of the geometry curriculum for sophomores, juniors, and seniors, in the plane and space. Features were chosen to discriminate data d corresponding to different classes r. Therefore, we can separate the feature space into zones where data mostly belongs to the same class. We will separate these zones with a linear boundary parameterized by x.

Suppose there are only two different classes, such as images of 2 and 7. Then the feature space must be separated into two zones. If there is only one feature, we compare $k = k_1$ to the value of a threshold x to define these two zones. If $k = (k_1, k_2)$ is in a plane, we divide this plane into two parts with a separating line. The equation of a non-horizontal line can be written $k_1 + ak_2 = b$. It is parameterized by $x = (a, b)$. If three features $k = (k_1, k_2, k_3)$ are used, defining a point in space, the separation is done with a plane equation $k_1 + ak_2 + ck_3 = b$. It is parameterized by $x = (a, b, c)$. The parameters of x modify the position and orientation of the separating line or plane. We will see that they are optimized during learning, to minimize the classification error calculated on the training examples.

These algorithms are linear classifiers. They separate the feature space k into two parts with a plane or line, whose equation can be rewritten with a dot product:

$$w.k - b = 0 \ .$$

The vector w is orthogonal to the line or plane. A point is on one side or the other of the plane or line depending on the sign of $w.k - b$, which thus defines the estimate $\hat{r}$ of the class.

Neural Networks Finding features that linearly separate all classes is the most challenging aspect of a data challenge. For complex problems, such as the CIFAR challenge shown in Figure 1, it is very difficult to find "by hand" features that do not produce many errors. A neural network can do better by also optimizing the choice of the list of features k. A neural network algorithm remains based on the calculation of dot products but also involves an activation function, which complicates the mathematical analysis. Studying a neural network provides students with an introduction to the most recent algorithmic approaches in artificial intelligence. However, their mathematical analysis is too complicated for high school and remains poorly understood.

A two-layer neural network first calculates each coordinate k_i of a feature k, with a weight vector w_i and a bias b_i:

$$k_i = s(w_i \,.\, x - b_i).$$

The neuron's "activation function" $s(t)$ can be a sigmoid or a rectifier $s(t) = \max(t, 0)$. The second layer performs a second linear separation $w.k - b$, where w is another weight vector and b is a bias. If two classes must be separated, only the sign of $w.k - b$ is retained. The parameters of the two layers of the network are the weights w_i and w and the biases b_i and b. They are optimized during learning, to reduce the estimation error on the training examples.

Using more than 2 layers can improve feature calculation. The most powerful neural networks can have several hundred layers, depending on the applications. For image recognition, large neural networks include several billion parameters, corresponding to the various weights and biases that calculate each layer from the previous one. For ChatGPT, the number of parameters is in the trillions.

3.2 Learning by Trial and Error: Optimization and Function Analysis

The learning phase optimizes the parameters x of the estimation algorithm to minimize the error between $\hat{r}$ and r. This involves both statistics and function analysis.

In a challenge, training data d with responses r is provided. The error $f(x)$ equals the average of the errors on all training examples between $\hat{r}$ and r. It is a function of the parameters of x. The simplest case studied in high school is when there is only one parameter, so $f(x)$ is a function of a single variable. For an image classification problem, the error equals the percentage of examples for which $\hat{r} = r$, for a fixed x:

$$f(x) = \frac{\text{Number of misclassified training images}}{\text{Number of training images}}.$$

Learning seeks a parameter $\hat{x}$ that minimizes the value of $f(x)$. At the heart of learning lies a mathematical and algorithmic optimization problem that must be solved efficiently with a computer. Learning by trial and error involves modifying x to reduce the error until the minimum is eventually reached. This can be done simply by calculating the error for many values of x and selecting $\hat{x}$ that minimizes $f(x)$. This involves plotting the function $f(x)$ and identifying its minimum. However, this approach is relatively inefficient and often too computationally expensive when there is a lot of data. Another approach is to gradually modify x to gradually reduce the error $f(x)$. This is done by calculating derivatives.

Learning in artificial intelligence involves many function analysis tools with derivative calculations. In senior year, this can include notions of convexity and convergence of iterative algorithms.

3.3 Generalization and Testing: Probability and Statistics

The danger of minimizing training error is learning a $\hat{x}$ that adapts too specifically to the training data. The worst case is simply memorizing the responses, like a student who learns the answers to math exercises "by heart." This gives a training error that is zero but becomes very large on new test data. As we have seen, the goal of learning is to generalize and therefore make few errors on examples not known in advance.

Test examples are provided by the challenge web platform. The test error is calculated for the best parameter $\hat{x}$ learned during training. For a classification problem, the test error equals the percentage of test examples for which $\hat{r} = r$, for $x = \hat{x}$:

$$f_{\text{test}}(\hat{x}) = \frac{\text{Number of misclassified test images}}{\text{Number of test images}}.$$

One cannot calculate this test error oneself because the platform does not provide the responses associated with the test data. This error is obtained by submitting the estimates $\hat{r}$ on the test data, and the platform (which has memorized the correct answers) displays the error (the score) on a leaderboard. If the test and training errors remain of the same order

$$f_{\text{test}}(\hat{x}) \approx f(\hat{x})$$

then this indicates good generalization on unknown examples.

Test and training errors have fluctuations because they are calculated by empirical averages on data assumed to be randomly taken, and therefore independent with the same distribution. The amplitude of these fluctuations is measured by the variance of these empirical averages. If the examples are taken randomly, this variance is inversely proportional to the number of training and test images. To ensure that learning on training examples does not suffer from these fluctuations, the number of training data must be sufficiently large relative to the number of parameters to be learned. These questions allow different levels of statistical problems to be studied, on the mean, variance, and the law of large numbers, which is at the heart of learning.

4 Teaching and Pedagogy

In practice, a difficulty in teaching mathematics through experimentation is ensuring that the experimentation is done within a limited time and fits into a balanced back-and-forth with the modeling and mathematical processing approach. With the time allocated to teaching mathematics in high school being greatly reduced, room must be made for abstraction and understanding the mathematics in the curriculum. These balances are difficult to establish and result from significant pedagogical and didactic work in the classrooms, conducted in co-development with the math labs [5] of the National Education of the Lille and Paris Academies.

Teaching Modules The teaching proposed by the MathAData team is organized into modules dedicated to different parts of the mathematics curriculum, in statistics, probability, geometry, function analysis, all including algebra. These modules can be used on different data challenges, depending on the interests of teachers and their students.

The first module includes an introduction to AI. An initial mathematical abstraction introduces the phases of estimation, learning, and testing, explained in section 3. This presentation is followed by a digital experiment on a challenge, such as image classification. Besides familiarizing themselves with the computer notebook and data, students must find a way to discriminate classes with one or two features they can calculate, which is a first step in mathematical modeling.

This introduction is integrated into a thematic module so that students immediately make the connection with the mathematics they are learning. Additional modules allow for further exploration of different aspects of the curriculum. The modules cover the following areas:

- *Statistics and Probability* for analyzing training error and generalization, with one feature: histograms, means, medians, variance, the law of large numbers, conditional probabilities.
- *Geometry* in the plane and space for linear classification: equations of lines in the plane, intersection of lines, bisectors, dot product, distance to a line, extension to space with plane equations, and distance between a point and a plane.
- *Function Analysis and Derivatives* for minimizing learning error by gradient descent: functions, minimum, derivative, sequence, sequence convergence, convexity.

An introductory module on two-layer neural networks will also be offered to students who have already completed the optimization and linear classification module. This module will have a strong component of digital experimentation. These modules can also provide an opportunity for students to undertake a more in-depth project that can be integrated into their high school oral exam.

Back-and-Forth Smoothing and properly framing the back-and-forth between computer experimentation and mathematical teaching is essential. Experimentation should be a source of questions, intuitions, and mathematical ideas rather than simple illustrations afterward. For simplicity, the teaching modules follow the same pedagogical sequence.

We start by explaining the motivation of the problem posed by the challenge, which typically has a component that goes far beyond this particular challenge, connected to more general problems. This can be image recognition, whale song recognition, or medical diagnosis. We then propose an approach that provides the main idea guiding the problem resolution in connection with the studied mathematics chapter in statistics, probability, geometry, or analysis. Computer experimentation allows students to manipulate and familiarize themselves with mathematical concepts intuitively, within the context of solving the challenge. This experience also allows them to test their ideas and ask new questions that are then addressed by mathematics. This is followed by a more formal presentation of the underlying mathematical concepts and practice with paper exercises proposed by the teachers.

For each module, the MathAData team provides a reference sheet with different possible sequences, as well as slides modifiable by teachers. The notebooks for computer experimentation are also divided into subparts, adaptable for each teacher's instruction. This does not require programming. Math exercises related to the challenge are also provided. All these pedagogical supports are co-developed with the math lab teachers [5]. These documents are available on our website *mathadata.fr*. We also organize training sessions to facilitate the use of these tools.

Digital Experimentation For data challenges, experimentation takes place in a computer environment that allows interaction with data and learning parameters without programming. This allows students to work more independently in mathematics classes.

For SNT or NSI classes, it is possible to introduce programming elements in Python, which opens up more flexibility. This experimentation then allows them to deepen their mastery of computer science and the Python language. Interfaces with the *challengedata.ens.fr* platform are provided to automate data access and easily submit results on test data.

The computer environment is in a Python Notebook, integrated into the National Education software *Basthon*. This allows working on the web browser without installing software. We also want to avoid that students, who often have very little programming experience, are not blocked by Python language bugs. For this, the programming is divided into small cells that implement functions of a few lines, for which we provide examples. These can be easily modified.

Scaling Up To scale nationally, it is necessary to help teachers get trained, develop, and provide pedagogical and computer supports adapted to the diversity of students. Scaling up is a well-known challenge for new educational initiatives. This is the focus of our collaboration with the math labs. It will also involve building a community of teachers who share their pedagogical resources. This work is initiated in collaboration with the National Education platform Capytale *https://capytale2.ac-paris.fr/web/accueil*.

Our goal is also to facilitate bridges between high schools and universities, with which we work for AI teaching through data challenges. This is particularly important to open perspectives for students who often ignore the considerable career opportunities in this field through mathematics.

5 Conclusion

For high schools, we see all the difficulties in implementing the natural and classic idea of teaching mathematics through questions posed by experimentation. The stakes are considerable for helping to better understand mathematics for a broader population of students and showing them that mathematics opens up future career opportunities. MathAData continues this pedagogical, computer, and classroom experimentation work, in co-development with the National Education math labs, because we believe the stakes are worth it.

6 References

1. *21 measures for teaching mathematics*, report by Cédric Villani and Charles Torossian, February 2018.
2. *Experimentation in mathematics*, Ciprelem Colloquium, June 2006.
3. *Experimental approach and mathematical learning*, INRP report, April 2007.
4. *Mathematics Laboratories in the AEFE network*, National Education report, 2021.