Data assimilation: mathematics for merging models and data

Matthias Morzfeld[1] • Sebastian Reich [2]

When you describe a physical process, e.g., the weather on Earth, or an engineered system, such as a self-driving car, you typically have two sources of information. You have a mathematical model and information obtained by collecting data. For example, you can measure temperature, or mount a lidar sensor on your car. To make the best predictions for the weather, or most effectively operate the self-driving car, you want to use both sources of information. Data assimilation describes the mathematical, numerical and computational framework for doing just that.

1 How we predict the future

The weather on Earth is interesting for many people. You might want to know if you need to bring an umbrella tomorrow. Safe operation of airports requires that one knows about a storm system that is heading north. The launch of a

[1] Matthias is supported by the Office of Naval Research under grant N00173-17-2-C003, by the National Science Foundation under grant DMS-1619630, and by the Alfred P. Sloan Foundation.
[2] Sebastian is supported by the Deutsche Forschungsgemeinschaft (DFG) under grant CRC 1114 "Scaling Cascades in Complex Systems" and grant CRC 1294 "Data Assimilation".

spacecraft also depends on weather conditions. And one must warn people in advance of severe weather such as flash floods, tornados, or even hurricanes. In fact, Earth's weather is so interesting that the one scientist almost everybody encounters on a daily basis is the meteorologist that presents your local weather forecast.

Since there is so much interest in weather, there are many people who study it using different (scientific) tools, and from different perspectives. For example, we measure temperature and humidity where we can, we use satellites to track storms, and airplanes or ships report the wind speeds they encounter. In addition, physicists and mathematicians work on developing mathematical models for the weather. Computational scientists work on how to use these models for simulations of Earth's weather. The weather forecast that you see on the evening news is a result of combining all that information.

Oftentimes, the combination of information from mathematical weather models and "observations" of the weather is done by using conditional probability. As a simple example, suppose that we roll a dice, but you keep your eyes closed. If I ask you what number is up, you might want to say 1, 2, 3, 4, 5, or 6, and all guesses are equally good. If I tell you that the number I see is larger than 3, then you might not want to guess 1, 2, or 3, since these numbers have become rather unlikely in view of the additional information I provided. Thus, your guess of the number should change because you obtained new information.

The situation is similar when you forecast the weather. You use a mathematical model to make a forecast. Then you obtain measurements of current weather, and you want to modify your model to account for this new information. For example, if you predicted sunshine for today, but today you observe rain, then you need to change your model and forecast for tomorrow in view of that observation. "Data assimilation" describes the mathematical foundations and numerical algorithms of how to make such changes to your model.

Weather prediction is one example where data assimilation is useful, but more examples can be found in almost every field in science and engineering. In robotics, for example in a self-driving car, data assimilation algorithms use sensor data to locate the car in a given map. The map itself must also be updated regularly based on the sensor measurements. For example, it is important that a pedestrian crossing the street becomes part of the map the car uses to navigate.

Another example is hydrology, where one studies the movement, distribution, and quality of water on Earth. When it rains, some of the rain flows across Earth's surface to end up in streams and rivers. Some of the water however penetrates the soil and travels underground. If you want to predict where this water ends up and how it flows under the surface, you will need to know what the structure under the surface looks like. To get an idea, you might drill

a few holes and make some measurements which you then want to combine with a mathematical model that describes the subsurface flow in between your measurements. The basic idea then is as before: you want to combine your model and data. You want to use data assimilation.

2 Mathematics of data assimilation

Data assimilation is done every day. In global numerical weather prediction, it is actually done every six hours. How? There are currently three main approaches to data assimilation.

1. Solve the actual problem.
2. Solve an optimization problem.
3. Solve a simplified problem.

An elegant way of performing data assimilation is to compute the conditional probabilities (see above) that describe the mathematical model in view of the data you collected (see [1, 6]). The conditional probabilities can be represented by a number of model "states", all of which are compatible with the data (up to the assumed errors). This approach is called "Monte Carlo", named after a casino in Monaco. The idea is that you can compute expected values or variances (see above) by averaging over the "samples" that you have.

As an example, consider again rolling a dice. The dice has six sides, with the numbers 1, 2, 3, 4, 5 and 6. The appearance of one side, say 1, is as likely as the appearance of any other side. That means that the probability of rolling a 1 is 1/6, and the probability of 2, 3, 4, 5, or 6 is also 1/6. The expected value is the sum of the numbers, multiplied by their probabilities:

$$E = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = 3.5$$

You can approximate this expected value by repeatedly rolling the dice, i.e., by drawing "samples". Suppose you draw 10 samples and you get the sequence shown in the first two rows of Table 1 (we rolled a dice to obtain this sequence). An approximation of the expected value is the sample average, i.e., the sum of the ten outcomes, divided by ten:

$$\hat{E}_{10} = \frac{1}{10} \cdot (5 + 3 + 6 + 2 + 1 + 2 + 5 + 1 + 3 + 1) = \frac{29}{10} = 2.9$$

The approximation gets better when you have more samples. An approximation based on 60 samples, which are also shown in Table 1, is given by the sum of all 60 outcomes, divided by 60:

$$\hat{E}_{60} = \frac{1}{60} \cdot (\text{sum of all outcomes}) \approx 3.42$$

Table 1: Rolling a dice 60 times

Roll-number:	1	2	3	4	5	6	7	8	9	10
Outcome:	5	3	6	2	1	2	5	1	3	1
Roll-number:	11	12	13	14	15	16	17	18	19	20
Outcome:	5	3	1	4	1	2	4	5	6	3
Roll-number:	21	22	23	24	25	26	27	28	29	30
Outcome:	5	2	1	3	1	6	3	1	5	3
Roll-number:	31	32	33	34	35	36	37	38	39	40
Outcome:	3	6	3	2	1	6	3	3	1	4
Roll-number:	41	42	43	44	45	46	47	48	49	50
Outcome:	5	6	4	3	1	6	4	3	6	6
Roll-number:	51	52	53	54	55	56	57	58	59	60
Outcome:	1	5	5	5	6	1	3	3	1	6

You can see that the approximation $\hat{E}_{60} \approx 3.42$ is now pretty close to the expected value $E = 3.5$. We encourage you to draw more samples, say 100, and you should get an answer even closer to 3.5. You can also tell ten of your friends to each draw 100 samples, then collect the results and average over all outcomes you and your friends observed. The result should be close to 3.5. If you were able to draw infinitely many samples, then you would obtain 3.5 exactly (see, e.g., [3]).

If you wanted to use Monte Carlo for data assimilation, then you would create ways of drawing samples from the conditional probabilities defined jointly by your mathematical model and the data. You can "use your imagination to do this" [3] and many scientists and mathematicians work together on finding new and effective ways to draw samples from probabilities that are not necessarily Gaussian. Designing such methods for data assimilation is very difficult, as explained in section 3 below. The reason is that doing data assimilation in this way becomes increasingly difficult, the larger the problem is. This is called the "curse of dimensionality", and, for this reason, data assimilation techniques of this kind are only used in relatively "small" problems.

The other two techniques mentioned above, i.e., optimization and simplifying the problem are often used in practice and even on very large and important problems such as numerical weather prediction. By an "optimization problem" we mean that you try to calibrate your model in a way that its output is as close

[3] This is a quote of somebody who creates such algorithms for a living. We collected this quote during a workshop at MFO on the topic of data assimilation.

as possible to the data you collected. You can quantify the mismatch of model output and data by a "cost function", and your "optimal" model output is the one that yields the smallest mismatch, which is also the smallest cost function. Typically, the cost function is related to a formulation of the data assimilation problem by conditional probabilities [7]. Optimization-based data assimilation algorithms are in use in operational numerical weather prediction, and some people say that progress in data assimilation is one of the main drivers for the increase in forecast skill over the past decades [2].

However, optimization can be cumbersome and difficult. The reason is that the implementation of these techniques in form of computer code can be tricky. There are problems for which this optimization approach is too complicated and difficult, and in these case you can make use of a very powerful method for solving difficult problems: just don't do it. Rather, you solve an easier problem, whose solution is not too far from the difficult problem you really should solve.

In data assimilation, this approach can be applied as follows. If the mathematical model is simple enough, the conditional probabilities that appear are also simple. By a simple mathematical model we mean one that is *linear*. Recall that a linear function $f(x)$ is one that satisfies $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$, where α and β are any numbers you want. Linear functions are usually easier to deal with than functions that are not linear. For example, you might find that solving the equation $f(x) = 0$ is easy if $f(x) = 2x + 1$ is linear, but difficult if $f(x) = x^2 + \tan(x)$ is nonlinear.

Linear models that appear in data assimilation satisfy similar conditions, and linear models are also "easy" to deal with. A linear model gives rise to "easy" conditional probabilities, which are described by "Gaussians". What makes Gaussian probabilities simple is that all you need to know about them are two numbers – the mean and the variance. [4] The mean is the average value you expect to encounter. The variance describes the variation around the mean that you should expect when you perform repeated experiments. In short, if you have a linear model and Gaussian probabilities, then you can perform data assimilation "easily" (see also the next section). Unfortunately, almost every physical process, or engineered system is not so well represented by a linear model, and probabilities that you observe are rarely Gaussian (there is no reason for them to be this particular way).

One way forward is to "linearize" your model. That means you replace the actual model by a linear version of it. The linearized model now gives rise to Gaussian probabilities and you are in business and you can "easily" solve this simplified problem. The solution you obtain can be very close to the solution of the nonlinear problem if the linearization captures most of the

[4] This is similar to linear functions, for which you only need to specify the slope and the offset.

dominant behaviors of your model. The Kalman filter [5] is the original data assimilation method for a class of linear problems, and versions of it, including the "extended" Kalman filter or ensemble Kalman filter [4] are used in many nonlinear problems.

3 Why is this difficult?

Data assimilation seems "easy". The problem is elegantly formulated in terms of conditional probabilities and there are three classes of methods to solve the problem. What is difficult about this?

One difficulty lies in the problem formulation. In order to set up the conditional probabilities, one relies on assumptions about the distribution of errors. For example, one must specify how "far" one expects the model outputs to be from the collected data. This is inherently difficult, because it requires statements, often in a precise mathematical sense, about things we truly do not know. How can we quantify the errors of a measured temperature to the output of a mathematical model, which is invented and not in any way related to temperature? Sometimes, one tries a few (or many) assumptions that describe such errors and then uses the one that leads to the "best" results. Here "best" usually refers to the set of assumptions that lead to the most useful forecasts and model outputs. Mathematical or computational convenience may also play a role. If one is not sure what assumptions to make, one first can try assumptions which simplify the problem.

Another difficulty is that the conditional probabilities are often far from "standard", i.e., far from probabilities that mathematicians or scientists understand well. This is mostly due to the fact that models of complicated physical or engineered processes are also complicated. Drawing samples from conditional probabilities that arise from complex mathematical models is a current topic of intense mathematical research.

A third difficulty is that many problems are "big". By big we mean that the number of variables that define the conditional probabilities is very large. Coming back to the example of numerical weather prediction, one can imagine that a model for the global weather requires that one specifies a lot of variables. In principle, you need to specify all meteorological quantities, e.g., the temperature, wind and humidity, at *all* locations on the globe. This is not possible because it would required infinitely many variables. What one does instead is to specify meteorological variables on a "grid", i.e., a fixed but large number of locations around the globe. In this way, a global mathematical weather model can have several hundred million variables. The number of data points is also large. Every six hours, two-ten million measurements are used to update the mathematical model.

The model itself, and the data assimilation method must then be implemented in the form of computer code. For a problem of the size of the global atmosphere, this requires careful algorithm design and coding. Today's data assimilation technology in numerical weather prediction are often based on optimization techniques or methods for simplified (linearized) problems. Many researchers also work on combining the three approaches.

In problems that are characterized by a large number of variables one must often fight the "curse of dimensionality" [5]. What is meant by this curse is that solving a problem becomes increasingly difficult as the number of variables, or "dimension", increases. And the rate with which the difficulty increases is very large.

You may recall the exponential function $f(x) = e^x$ ($e \approx 2.718$ is Euler number), which grows very quickly with x. For example, $e^1 \approx 2.718$, $e^2 \approx 7.389$, $e^3 \approx 20.086$. The "curse of dimensionality" is that if x is the number of variables, then the difficulty level is e^x. As an example, in Monte Carlo, the difficulty level may by described by the number of samples required to compute an expected value with a given accuracy, e.g., with 10% error (in the dice example above, we use 60 samples). For the data assimilation of numerical weather prediction, the number of variables is 200 million (or more). The difficulty level, as measured by the number of samples required is then $e^{200 \cdot 10^6}$ which is much larger than the number of atoms in the universe (estimated to be between 10^{78} to 10^{82}). It is impossible to draw that many samples, even on today's powerful supercomputers. There is currently a lot of interest to somehow overcome the curse of dimensionality, especially in Monte Carlo sampling and data assimilation, but at this time it is unclear how to do it. In a future Snapshot, you might read about the solution of how to overcome these difficulties without making overly simplifying assumptions.

4 Summary

Data assimilation means to merge a mathematical model with information obtained from data (measurements). Many scientific and engineering problems require data assimilation. The list of problems where data assimilation is useful is very long but includes numerical weather prediction, hydrology, personalized medicine, cognitive science, and robotics. We have discussed a few of these applications but focussed on numerical weather prediction. We have also introduced the three main approaches to solving data assimilation problems, and explained why solving data assimilation problems is difficult.

The difficulties one encounters are of course different from problem to problem.

[5] This is true for many problems, not just data assimilation.

You can imagine that solving a data assimilation problem in the context of numerical weather prediction is very different from solving a data assimilation problem related to a self-driving car. In fact every scientific or engineering problem has its very own challenges and characteristics. Challenges can include an immense number of variables, or probabilities that are far from those we understand well. Each data assimilation problem also comes with its own mathematical model and data types, and the computational architecture you can use to solve it may also vary. For example, you are probably given a large computer when you make a weather forecast, but to operate a self-driving car, you have access to much less computational power.

In view of data assimilation's wide use in different applications, it is surprising that all of these problems have essentially the same formulation in terms of conditional probabilities and three main avenues to success (Monte Carlo methods, optimization and linearization). It is up to the mathematician, engineer and scientist to combine these methods to from a suitable recipe for the solution of the data assimilation problem at hand. In this context, mathematics is particularly useful to identify "classes" of problems that consist of problems which are characterized by a certain set of common specifications. Once a problem class is determined, one can look for suitable algorithms for the solution of all problems within that class.

References

[1] M. Asch, M Bocquet, and M. Nodet, *Data assimilation. methods, algorithms and applications*, SIAM, 2017.

[2] P. Bauer, A. Thorpe, and G. Brunet, *The quiet revolution of numerical weather prediction*, Nature **525** (2015), 47–55.

[3] A.J. Chorin and O.H. Hald, *Stochastic tools in mathematics and science*, third ed., Springer, 2013.

[4] G. Evensen, *Data assimilation: the ensemble Kalman filter*, Springer, 2006.

[5] R.E. Kalman, *A new approach to linear filtering and prediction problems*, Journal of Basic Engineering **82** (1960), no. 1, 35–45.

[6] K.J.H. Law, A. Stuart, and K. Zygalakis, *Data assimilation: a mathematical introduction*, Springer, 2015.

[7] O. Talagrand and P. Courtier, *Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory*, Quarterly Journal of the Royal Meteorological Society **113** (1987), no. 478, 1311–1328.

Matthias Morzfeld *is an assistant professor of mathematics at the University of Arizona (USA).*

Sebastian Reich *is a professor of mathematics at the University of Potsdam (Germany) and at the University of Reading (UK).*

Mathematical subjects
will be filled out by the editors

Connections to other fields
will be filled out by the editors

License
will be filled out by the editors

DOI
will be filled out by the editors

Snapshots of modern mathematics from Oberwolfach provide exciting insights into current mathematical research. They are written by participants in the scientific program of the Mathematisches Forschungsinstitut Oberwolfach (MFO). The snapshot project is designed to promote the understanding and appreciation of modern mathematics and mathematical research in the interested public worldwide. All snapshots are published in cooperation with the IMAGINARY platform and can be found on www.imaginary.org/snapshots and on www.mfo.de/snapshots.

Junior Editor
will be filled out by the editors
junior-editors@mfo.de

Senior Editor
Carla Cederbaum
senior-editor@mfo.de

Mathematisches Forschungsinstitut
Oberwolfach gGmbH
Schwarzwaldstr. 9–11
77709 Oberwolfach
Germany

Director
Gerhard Huisken

Mathematisches
Forschungsinstitut
Oberwolfach

Member of Leibniz Association

IMAGINARY
open mathematics