Trusting a black box: explaining complex simulation outcomes using LIME

Christoforos Anagnostopoulos
Imperial College
London, UK
christoforos.anagnostopoulos06@imperial.ac.uk

Stefano Romano
Improbable
London, UK
stefanoromano@improbable.io

ABSTRACT

The field of Artificial Intelligence (AI) has recently been suffering an 'interpretability crisis'. Black-box techniques like deep learning produce impressively accurate predictions, but often fail to offer any human intelligible explanation. This makes it hard to establish their safety and fitness-of-purpose in highly regulated or safety-critical domains. In the adjacent field of modelling and simulation, this challenge is not new: complex simulations generate emergent outputs in ways that are often opaque to the user, and sensitive to initial parameter settings or details of their initial state such as the exact placement of units on a map. This can be broadly quantified via classical techniques such as sensitivity analysis, which however fail to provide a true explanation of simulation outputs in terms of understandable features of the input space. As a response to the interpretability crisis in AI, the new field of Explainable AI (XAI) has emerged in recent years. Techniques like Locally Interpretable Model-agnostic Explanations (LIME) enable powerful post-hoc analysis of predictions that provides an intuition for the model's logic. This is achieved by considering how small, local changes to the input configuration affect the response, capturing the resulting dependencies using interpretable statistical techniques, and reporting them in an intuitive graphical user interface. This closes the loop between the model and the user, allowing the user not only to build trust in the model, but also to actively improve it by identifying blind spots and misconceptions that are evident to a human expert. In a series of examples from the social sciences and epidemiology, including an influential model in the Covid-19 response policies in the UK, we use LIME to coherently trace emergent patterns back to the input space.

ABOUT THE AUTHORS

Christoforos Anagnostopoulos, PhD. holds degrees in mathematics and theoretical computer science and following a PhD in streaming data analysis was a lecturer in statistics at Imperial College London where he taught graphical modelling. After five years of running a niche consulting startup in the area of cyber security he joined Improbable as a Senior Principal Scientist in their modelling and simulation arm. He is now back at Imperial College London where he teaches ethics of data science and AI.

Stefano Romano, PhD. is an Engineering Manager in the Complexity Research team at Improbable. He holds a PhD in mathematical physics from International School for Advanced Studies (Trieste, Italy). His recent research interests are in machine learning, data science and complex systems.

Trusting a black box: explaining complex simulation outcomes using LIME

Christoforos Anagnostopoulos
Imperial College
London, UK
christoforos.anagnostopoulos06@imperial.ac.uk

Stefano Romano
Improbable
London, UK
stefanoromano@improbable.io

INTRODUCTION

A confluence of technological advances has led to increased adoption of computational models for decision support in multiple domains, including policy making, clinical diagnosis, education, as well as military training and planning. In part, this is driven by recent advances in artificial intelligence in areas such as clinical diagnosis (Hosny et al., 2018), but it is also a result of a long-term trend in more widespread adoption of mathematical modelling in public policy. Countries like the United Kingdom have taken official positions in favour of model-based policy making while simultaneously insisting on best practices, as outlined for example in the Aqua Book (HMG, 2015). In particular, model-driven decisions about public health interventions have increasingly gained foothold in government (Boden & McKendrick, 2017). This was brought to prominence during the recent Covid-19 pandemic, which at the time of writing has already extolled a terrible cost in lost lives (World Health Organisation, 2020) and livelihoods (European Council, 2020).

The term "computational model" itself is overloaded, encompassing a broad diversity of different modelling techniques (Calder et al., 2018, section 5.1). For the purposes of this paper, it suffices to draw the distinction between *empirical* (or *data-driven*) modelling, whose internal structure and parameter values are almost fully determined by the data using some form of optimisation, versus *substantive* modelling, whose internal structure is almost fully determined by human expertise, scientific or otherwise (Hand, 2013).

In the case of empirical modelling, recent years have seen an explosion of complexity in internal model structure, with the introduction of end-to-end deep neural networks (Amodei et al., 2016; Bach et al., 2015). Deep neural networks are computational representations of highly non-linear mathematical functions, whose parameters are automatically trained using labelled data examples. They are referred to as end-to-end when the input to the neural network comprises raw, often unstructured data (such as images), without any attempt to summarise it into useful features with the aid of a human expert.

This approach has been proven extremely successful at extracting meaningful patterns from the data that the model observed, which can then be applied to new, unobserved examples, and often result in super-human performance in classification tasks such as recognising the content of an image. On the other hand, the explosion in the complexity of internal structure has posed what has come to be known as an *interpretability crisis*: human end users that receive recommendations by decision-support systems based on deep learning are unable to interpret the reasons for these recommendations (Ahmad et al., 2018). As a result, they are unlikely to trust them, especially when these recommendations run counter to human intuition. This is unfortunate, as it is really in this latter circumstance that we expect decision support systems to add the most value, that is, in situations where the computational model is able to surpass human intuition in its generalisation ability. In response, this has triggered a novel area of scientific research, known as *explainability*. Explainability is a more relaxed requirement than interpretability, in that it allows for situations where the computational model is too complex to be understood by a human, as long as there is some mechanism for the model to reliably *explain* its recommendations, i.e. offer an interpretable approximate summary of any given decision. This helps bridge the gap of trust introduced by the inscrutability of black-box algorithms. A number of techniques have appeared in the literature that treat the computational model as an input-output black-box. Locally Interpretable Model-agnostic Explanations (LIME) (M. T. Ribeiro et al., 2016; M. Ribeiro et al., 2019) is an example of such a method and one we will focus on in this work, as its model-agnostic nature renders it applicable to computational models of any kind, not just empirical models.

Armed with this generic explainability tool, we turn our attention to substantive models, and in particular, a class of models known as agent-based, or individual-based models (Epstein, 1999; Gilbert & Terna, 2000; Epstein, 2006; DeAngelis, 2018), or even sometimes as micro-simulators (Nagel & Rickert, 2001). For brevity, we will henceforth refer to all such models as agent-based, or ABMs. What ABMs have in common is that they represent in a computer program complex interactions between a large number of individual agents. These could be soldiers on a virtual battlefield (Cioppa et al., 2004), pedestrians in traffic simulation (Gipps & Marksjö, 1985), mosquitos in a model of malaria transmission (Gu et al., 2003), or civilians joining a riot (Epstein, 2002). The ability to represent individual agents gives extensive power to the modeller to stack up complex behavioural logic at the agent level that depends on an individual's history, other agents, as well as a richly represented environment. This makes ABMs particularly expressive as repositories of substantive knowledge, often parameterised in terms of quantities that can be fixed using data or expert opinion: for example, an ABM modelling epidemic spread will typically involve average household size as an input parameter.

In the Covid-19 pandemic, ABMs were brought to the forefront of public debate as they were extensively used by public health officials to determine the relative merits of different non-pharmaceutical interventions (i.e., interventions other than treatment or vaccination), such as *lockdown*, where a certain region's mobility is drastically reduced via compulsory measures, and, at the other extreme, *voluntary social distancing*, where civilians are instead advised, but not obliged, to reduce their social contacts as much as they can. For certain interventions in particular, such as *digital contact tracing*, which involves the use of a smartphone app to facilitate in tracing contacts of infected individuals by public health officials, it was claimed (Ferretti et al., 2020) that ABMs were in fact the only class of models capable of forecasting the likely impact of this intervention to the epidemic.

Despite their substantive nature, and the interpretability of individual sub-components of an ABM (such as, say, household composition), the coupling together of numerous types of agents with different types of behaviours amounts to an unwieldy computational model that is hard to validate (Windrum et al., 2007) or to intuitively grasp in full. Moreover, just like in deep learning, an accurate ABM is most useful as a companion to human decision making when it surpasses our intuition in producing a surprising or counter-intuitive answer—with the troubled term *emergent behaviour* often used to capture such hard-to-predict aggregate phenomena arising out of a complex system of relatively simple agents (Epstein, 1999)—which immediately raises an interpretability challenge. This is compounded by the fact that ABMs are also highly non-linear functions of numerous input parameters, and can be very sensitive to changes in these parameters, as well as to the initial states of the agents in the simulation[1]. In safety critical or public policy areas, these interpretability considerations are paramount, and were at the root of significant public concern and confusion in the Covid-19 pandemic, too.

To conclude, we recognise that an ABM from a user's perspective suffers the same lack of transparency and interpretability as a deep learning model, and explore whether it could therefore also benefit from similar explainability techniques.

The rest of the paper is structured as follows. In Section 2, we lay out the methodological framework we are proposing, starting in Section 2.1 by a description of LIME as originally proposed for explainability of machine learning models. In Section 2.2, we proceed to offer a detailed description of how ABMs can be recast as a generic input-output black box, rendering them amenable to LIME analysis, and in Section 2.3 we take a detour into prior art that does not belong in the explainability literature but is in fact related, such as, notably, sensitivity analysis (Saltelli et al., 2004). Section 3 is a brief introduction to the specific ABMs we will be exploring in this paper. Section 4 contains our experimental results, and finally we conclude with a discussion and a description of future work in Section 5.

FRAMEWORK

We represent an ABM or indeed any type of simulator as a pseudo-random program $\mathcal{M}$ that takes as input a vector of configuration parameters $\vec{\theta} = (\theta_1, \ldots, \theta_p)$, as well as an initial state for all the agents in the simulation, z_0, and proceeds to run a forward simulation of a certain system. When the simulation ends, which is typically a fixed time duration T (where time here is taken to mean time as represented inside the model, rather than the physical time it

[1] The initial state configuration is especially problematic, as its dimensionality in principle scales linearly with the number of agents.

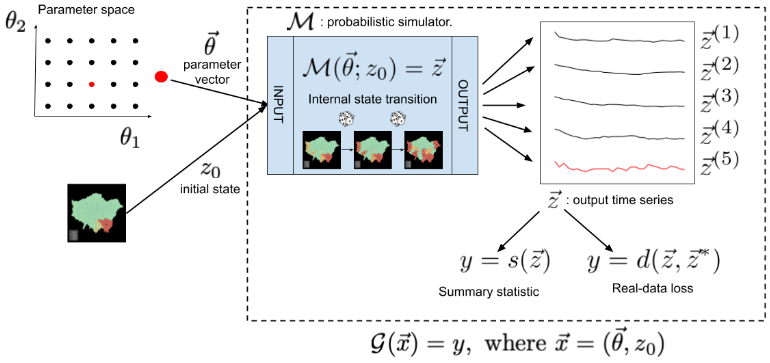
takes to run the simulation), it produces statistics that capture the outcome of the simulation. These might represent the end state of the simulation (for example, total number of deaths before herd immunity in an epidemic model), or they might summarily describe the track followed by the simulation from time 0 to time T (for example, the time series of deaths from the start to the end of the epidemic). We can represent most situations in some generality by assuming the latter, namely that $\mathcal{M}\left(\vec{\theta}, z_0\right) = (z_0, z_1, \dots, z_T)$, where we add the initial state z_0 as part of the output for convenience.

We can then additionally consider a *summary statistic* function that reduces $\vec{z} = (z_0, z_1, \dots, z_T)$ to a single number, $y = s(\vec{z})$. For example, if we fix the unit of time implicit in the duration T to be one day, and z_i be the daily deaths during a simulated epidemic, we can reduce that to the total death count y by simply summing over the curve, so $y = s(\vec{z}) = \sum_{i=0}^{T} z_i$. A wide variety of different summary statistics might be interesting to consider, such as the peak of the curve (its location in time or height) or the amount of time for which the daily cases exceed the capacity of a given national health system.

It is worth noting that although z_i here suggests discrete, regular time (i.e., we record output from the simulator once a day), the simulator itself might operate instead in a discrete-event formalism, that is, one where the agents are allowed to act asynchronously at times that are usually drawn from an appropriate probability distribution (Fishman, 2013). Regardless of the internals of the simulator, we choose in this work to represent its output as a regular time series, which we refer to as a simulation *trace*.

It is useful for the purposes of our study to encapsulate the combination of the simulator $\mathcal{M}(.,.)$ and the summary statistic function $s(.)$ into a single function $y = \mathcal{G}(\vec{x})$, where $\vec{x} = \left(\vec{\theta}, z_0\right)$. As a final important detail, we note that many ABMs are probabilistic, in that they entail draws from random distributions to determine components of behaviour or state transitions. As a result, multiple runs from the same simulator will produce varied output. It is also worth clarifying that this can be controlled from an implementation perspective by fixing the seed of the random number generator in single-threaded execution. This aspect of the simulator can be captured by the notation $y \sim \mathcal{G}(\vec{x})$ where $\mathcal{G}$ now represents a probability distribution rather than a function.

Figure 1 illustrates the above formalism visually. On the top left we see a grid of candidate parameter values for the configuration parameter vector $\vec{\theta}$. When exploring the behaviour of an ABM, grid search on the input space is often resorted to, or variations thereof (Cannon et al., 2020). On the bottom left we see a visual representation of what an initial state specification might look like. In this case, we see a heatmap of London – for concreteness, imagine we are looking at a civil unrest simulation and each region is colored according to intensity of riots at the start of the simulation[2]. In the centre of the figure we can see a representation of the internal structure of the simulator, proceding here for visual clarity in discrete state transitions, possibly involving calls to random number generators, represented by dice. Multiple runs of the simulator for the same input $\vec{x} = \left(\vec{\theta}, z_0\right)$ will then produce possibly different output traces, denoted by $\overrightarrow{z^{(1)}}, \dots, \overrightarrow{z^{(N)}}$, that can be respectively summarised as $y^{(1)}, \dots, y^{(N)}$, with $y^{(i)} = s\left(\overrightarrow{z^{(i)}}\right)$. Notationally, we use superscripted parentheses to indicate multiple samples from a variable, and subscripts to indicate components of a multi-dimensional vector.


Parameter space
θ_2
θ_1
$\vec{\theta}$
parameter vector
z_0
initial state
$\mathcal{M}$: probabilistic simulator.
INPUT
$\mathcal{M}(\vec{\theta}; z_0) = \vec{z}$
Internal state transition
OUTPUT
$\vec{z}^{(1)}$
$\vec{z}^{(2)}$
$\vec{z}^{(3)}$
$\vec{z}^{(4)}$
$\vec{z}^{(5)}$
$\vec{z}$: output time series
$y = s(\vec{z})$
Summary statistic
$y = d(\vec{z}, \vec{z}^{*})$
Real-data loss
$\mathcal{G}(\vec{x}) = y, \text{ where } \vec{x} = (\vec{\theta}, z_0)$

[2] This representation is useful to us later, but it is worth noting that not all ABMs will have a natural spatial mesh representation of their initial state.

Figure 1. A LIME explainer on the output of the Covid-19 contact tracing model.

This process of generating multiple output traces for a given input configuration, sometimes referred to as *Monte Carlo* analysis, offers a simple approximation to the mathematical expectation under the distribution $\mathcal{G}$:

$$E_{\mathcal{G}}(f(y)) \approx \frac{1}{N}\sum_{i=1}^{N} f(y^{(i)}),$$

for an arbitrary function *f.*

Local Interpretable Model-Agnostic Explanations (LIME)

Explainability techniques typically operate in the context of *supervised classification* (see for example Murty & Devi, 2011), wherein the objective is to use a training dataset of labelled examples $(X^{(i)}, y^{(i)})_{i=1:N}$, where $X^{(i)}$ is the example in question (e.g. an image) and $y^{(i)}$ the respective true label (e.g. "cat"), to automatically identify a mapping $f_{\psi}(X) = \hat{y}$ that is able to offer a prediction $\hat{y}$ of the label of possibly unseen examples X. The mapping f_{ψ} is typically parameterised by a vector ψ, which is effectively set to the value $\tilde{\psi}$ that maximises predictive accuracy in the training data, possibly subject to regularisation constraints. As a result, a trained classifier is an object $f: X \mapsto \hat{y}$. For the purposes of LIME, this object can be thought of as a black-box with input X and output y—its parametric form, the training data and the training methodology are all irrelevant, by virtue of the model-agnostic property of LIME and related techniques.

For the purpose of this paper, we can assume that the input X is a numeric vector. More generally, LIME can handle cases where X includes categorical variables, or consists of unstructured text or image data. In all cases, LIME requires the input to be converted into to a binary indicator vector $x' \in [0, 1]^d$, where each $x_i{}'$ indicates the presence or absence of an interpretable *feature*. The construction of the features can be user-specified, but by default numeric variables will be binned (either via bins of fixed lengths or via quantile bins) and categorical variables will be one-hot-encoded. Images and text are treated a little more idiomatically, and though they are an interesting direction for future work, are not discussed here. We henceforth refer to the transformation of x into x' as ϕ, and assume a reconstruction ϕ^{-1} exists[3] so that for any v' we can identify a v such that $\phi(v) = v'$.

LIME can be derived from an abstract argument, as demonstrated in part in the original paper (M. T. Ribeiro et al., 2016) and in full in (Lundberg & Lee, 2017), but here we start instead with a concrete example, to emphasise how intuitive the underlying algorithm is. The basic premise is that the function $f(x) = y$ is impossible to understand globally, but can be approximated locally by a linear model. This is achieved by sampling a number of points from the neighborhood of x, say, $x_1, \dots, x_N$, obtaining their labels $y_1, \dots, y_N$, and then mapping the x_i onto their interpretable versions, $x_i' = \phi(x_i)$.

The perturbation itself is produced by mapping x onto its interpretable, binary version x', as described in the previous paragraph, then perturbing x' simply by flipping a (uniformly) random number of bits, and then mapping the perturbed points back to the original space:

- Given x, $f(.)$, $\phi(.)$ and $\phi^{-1}(.)$:
- Obtain interpretable version $x' = \phi(x)$.
- Sample N datapoints $x_1', \dots, x_N'$ from the neighborhood of x'.
- Apply ϕ^{-1} to obtain their counterparts $x_i = \phi^{-1}(x_i')$.
- Get their labels $y_i = f(x_i)$
- Fit a linear model on the data $(x_i', y_i)_{i=1:N}$.

[3] For binned numeric variables, an exact inverse is not available. In the implementation of LIME used in this paper, binned variables are reconstructed by sampling from a truncated normal distribution inside the bin, with mean and variance derived from the training data.

The linear model that this process returns is known as an *explanation*. LIME employs a sparse linear model to ensure that the explanation only comprises the most significant features (i.e., components of x'). Sparsity can be achieved in linear models in a variety of ways, but the most frequent one is to modify the ordinary least squares solution by a adding a penalty term that involves the absolute values of the regression coefficients. This technique was popularised by Tibshirani (1996) and is known as Least Absolute Shrinkage and Selection Operator (LASSO).

The above describes completely the way in which LIME can offer explanations of a single datapoint, x. However, the user might also be interested in forming a picture of the model as a whole. LIME can cater for that by using heuristics to select a number of interesting points in parameter space to look at. In Figure 1 we present a grid search, but in practice this is likely wasteful, because some areas in parameter space might have much more variability than others, so that a grid might be an inefficient way to summarise the entire model via local explanations. Ribeiro et al. (2019) propose what they refer to as a *submodular pick* algorithm, which, starting from a set of points in the parameter space $\{x_1, \dots, x_N\}$, and a budget of model evaluations B, produces local explanations in a greedy fashion, attempting to maximise the so-called *coverage* of the feature space: recall that due to sparsity, each explanation only involves a small number of features, so an attempt is made to expose the user to all possible features.

Related Work

The modelling and simulation ecosystem has been grappling with the problem of exploring the breadth of simulation outcomes and their dependence on configuration parameters and initial values for a long time, independently of the explainability literature. In this section, we describe in brief two areas in particular: sensitivity analysis, which, like LIME, uses simple statistical models to understand the dependence of the simulation output to its input; and, scenario analysis, which proposes useful criteria for which outcomes among the space of all possible outcomes might be worth focusing on.

Sensitivity analysis

Solutions to this problem often fall under the guise of *sensitivity analysis* (SA) (Saltelli et al., 2004). Many SA techniques assume that the model under scrutiny is mathematically tractable so that gradient-based techniques can be deployed to get explicit measures of sensitivity to different parameters, but others take a similar approach to ours, treating the model as a black-box and producing Monte Carlo samples from it by evaluating its answer on a set of parameter values, usually explored at random or using grid search, and then fitting a linear model on the resulting dataset. Monte-Carlo based SA techniques have significant similarities with our proposed approach. For example, in Regional Sensitivity Analysis (RSA) (Hornberger & Spear, 1981), the output of the scientific model is similarly reduced to a binary summary statistic, as we propose in the previous section, whereas in more recent RSA techniques (Pappenberger et al., 2006) flexible tree-based models are used to assess sensitivity, which amounts to a locality assumption. Similarly, significance testing-based techniques (Hamby, 1995) carve up the space in multiple regions and attempt to ascertain whether the output is significantly different between such regions.

The main differences of LIME over the SA literature can be summarised as follows:

- LIME's feature extraction step ensures that the local explanations are always linear models of the same form: LASSO regression on a binary input vector. As a result, the output LIME does not require statistical expertise, in contrast to the usual graphical tools employed in SA, and is much more user-friendly and intuitive, which means it can be available at the point of need as a general-purpose explanation of the model output, rather than just during model development and validation.
- LIME applies to images and text, not just numeric and categorical input variables.
- Local SA techniques typically attempt to break the parameter space into regions that exhibit similar output behaviour. In contrast, local explanations are concerned with a single location in parameter space, not a contiguous region.
- SA is mostly concerned with explaining variation, whereas LIME is also interested on direction: which features is it that made the model think that y is the right answer, and which worked against it?

It is also worth pointing out that sensitivity analysis often complements *calibration*, that is the attempt to set the parameters in a way that results in the most faithful representation of the real-world system in question. Calibration might happen via expert or data-driven specifications of the input parameters directly, or indirectly, via optimising a loss function comparing the actual real-world time series to the model output (see Figure 1, right hand side). Calibration and sensitivity analysis work hand-in-hand (Ratto et al., 2001): the parameters to which the model is most sensitive are the best candidates for calibration; and, conversely, any parameters that we are unable to or lack the knowhow to calibrate should be ones that the model is not particularly sensitive to. LIME can also interact fruitfully with calibration. If the model $\mathcal{M}(\vec{\theta}; z_0)$ has been calibrated to a value $\hat{\theta}$, LIME can then offer an explanation of the model output by fitting a linear model precisely on the neighborhood of the calibrated values.

Sensitivity analysis

Just like global sensitivity analysis can be turned local by splitting the parameter space into regions, an inherently local technique like LIME can be turned into a global one using the submodular pick algorithm of Section 3. In scenario planning (Chermack et al., 2001), a common approach is to focus on outcomes instead. Commonly, the planner will focus on *best-case*, *worst-case*, and *most likely* scenarios (Taylor et al., 2017). Converting the raw simulation output, denoted in Figure 1 as $\vec{z}$, into a loss can be made difficult by the need to balance different considerations against each other. For example, the total number of deaths is typically the loss function that epidemiologists want to minimise during in epidemic, though other considerations might eventually become important, like the excess unemployment rate caused by different interventions. We do not discuss such issues further, and instead assume that the summary statistic $s(\vec{z})$ can also act as a loss function.

CASE STUDIES

In this section, we briefly introduce the two models of interest for this paper: the civil unrest model introduced by Epstein (2002), and the Covid-19 epidemic model produced by Ferretti et al. (2020).

The Epstein Civil Unrest Model

Epstein (2002) describes an ABM of rioting behaviour. The population comprises N_c civilians and N_p police officers, each occupying a slot in a square lattice world, represented as a $L \times L$ grid, and move left, right, up or down at random on every iteration. Civilians can either be actively rioting, or not. Whether or not the ith cilivian will be active or inactive at time t is denoted by A_{it}, given by

$$A_{it} = \begin{cases} True, & if \;\; H_i \cdot (1-L) - p \cdot R_i > T \\ False, & \quad otherwise \end{cases} \quad , \quad p = 1 - e^{k \cdot \frac{p_i(t)+1}{r_i(t)+1}}$$

where T is an activation threshold, H_i is a measure of how much hardship civilian i is experiencing, L is a global legitimacy measure which represents how much faith civilians have in the government, R_i is the risk-appetite of the ith civilian, k is a “temperature”-like constant. $p_i(t), r_i(t)$ denote the number of police officers and rioters near civilian i at time t respectively, where "near" means within a Euclidean distance d_n of the civilian's location. Finally, on every iteration, a police officer can capture a civilian who is rioting and is within the officer's field of sight (within Euclidean distance d_v), in which case the civilian is removed ("jailed") for n_J days. The output of the model is then a time series of active rioters over time, and the intention is to minimise "eruptions" of civil unrest, where frustration builds up and is released at some random point resulting in a large-scale riot that overwhelms the police force, decreasing the risk of arrest, which leads to more riots. As a result, the output is $z_t = \sum_i A_{it}$ and our summary statistic its peak, $y = s(\vec{z}) = \max_t z_t$. Inspecting these configuration parameters suggests that rioting should *increase* as:

- the following go *up*: hardship H_i, risk-appetite R_i, number of civilians N_c, “temperature” k.
- and the following go *down*: legitimacy L, activation threshold T, police vision d_v, jail time n_J.

The distance d_n, used by civilian agents to compute “nearby” rioters and police officers, is harder to analyse upon mere inspection, as well as any role played by initial configurations of police versus civilians on the map. To explore the latter, we experiment with a non-uniform placing of police officers in the map, guided by a mixture of exponential kernels, whose radius is controlled by an additional configuration parameter (denoted `police_init_kernel_radius` in the following). The larger the radius, the more uniform the placement of the units, whereas for smaller radii, the units are placed in concentrated clusters, with low coverage in between.

The COVID-19 Contact Tracing Model

Ferretti et al. (2020) describe an ABM for understanding the effect of different non-pharmaceutical interventions on the spread of the Covid-19 pandemic. The model places agents on a randomly generated network, and simulates a number of contacts per day for each different age demographic, in the household, in the community and in workplaces/schools. Infections then occur as a result of a fraction of these contacts, some of which proceed to hospitalisations, and some of those to death. The richness of the ABM allows the same model to simulate a number of non-pharmaceutical interventions such as school closures, lockdown, digital contact tracing, and more. The code for the model has been open-sourced[4]. The ABM is under active development, and at the time of writing it has 179 configuration parameters[5], among which a good number are calibrated using demographic data. These fall roughly in three categories: parameters governing the infection dynamic, the disease progression and the effect of interventions. For the purposes of this presentation we focused on the first category (infection dynamic), and restricted to parameters for which a direct method of estimation was not provided in the model documentation[6], listed in the table below. This makes them natural candidates for explanation, to build a quantitative picture of how the uncertainty in these values translates to model outcomes in key scenarios.

Table 1. Parameters and their value range for the COVID-19 model.

	Default value	Lower bound	Upper bound
`mean_infectious_period`	6.00	3.00	12.00
`asymptomatic_infectious_factor`	0.29	0.14	0.57
`mild_infectious_factor`	0.48	0.24	0.96
`relative_transmission_household`	2.00	1.00	4.00
`relative_transmission_occupation`	1.00	0.50	2.00
`relative_transmission_random`	1.00	0.50	2.00

We refer the reader to the model documentation for a detailed description of these parameters, and only provide a short summary here: `mean_infectious_period` governs the infectiousness profile of infected individuals over time; `asymptomatic_infectious_factor` and `mild_infectious_factor` discount infectiousness for individuals with no symptoms and mild symptoms respectively; the `relative_transmission` coefficients determine the rate of infection between individuals in the same household, in the same workplace/school and randomly in the community.

COMPUTATIONAL EXPERIMENTS

We now describe our suite of computational experiments. We make use of Python package `lime`[7] from the original LIME authors (M. Ribeiro et al., 2019). We begin by specifying a range for each parameter in θ that we wish to explore, and a specified number of bins to use per parameter. For simplicity, we start by producing a number of Monte Carlo runs by sampling from the parameter space according to some pre-specified sampling scheme. This could be an orthogonal grid, or a uniform sampler. We then pre-evaluate the model for each parameter setting sampled, and select the best-performing example, the worst-performing example, and an average-case example, which we proceed to run a LIME explainer on. Recall that the LIME explainer will request further model evaluations for each example that is being explained. In cases where model evaluations are very expensive, a surrogate machine learning regressor can be used in the model's place, by pre-training it on the set of examples mentioned above, which should in such a case accordingly be made larger. This amortizes the cost of model evaluations, and can make the explainer available at the point of need, even on very lightweight computational infrastructure.

[4] https://github.com/BDI-pathogens/OpenABM-Covid19
[5] https://github.com/BDI-pathogens/OpenABM-Covid19/blob/master/tests/data/baseline_parameters_transpose.csv
[6] https://github.com/BDI-pathogens/OpenABM-Covid19/blob/master/documentation/covid19_model.pdf, Table 6
[7] https://github.com/marcotcr/lime

Additionally, we produce a global explainer, to allow us to compare to global sensitivity analysis. Rather than using a bespoke global SA algorithm, we run LASSO again on the examples mentioned above, to maintain visual consistency and make it easy to compare with LIME. In Figures 2 and 3 we see the main user report generated by LIME, containing results from the following analyses: global, best-case, worst-case, and average-case respectively. Each subplot contains one horizontal bar chart for each feature, for the three features deemed most important by the explainer in question. Recall that features here are binned version of original parameters. For all but the global explanation, we employ the following structure in the y-axis label:

```
bin_lower_bound < feature_name < bin_upper_bound [global_min < value < global_max]
```

where `value` is the numerical value of original (unbinned) parameter for the scenario explained. So, for example, a label that might read "$k \leq 10$ $[5 < 6 < 20]$", which indicates that the example in question had a value of $k = 6$, whereas the global range is the interval $[5, 20]$, and the feature in question is that k should be ≤ 10. The value of the LIME score itself is indicated in green if positive and red if negative. Positive values suggest that the presence of the feature is conducive to a good outcome, and its absence to a bad outcome.

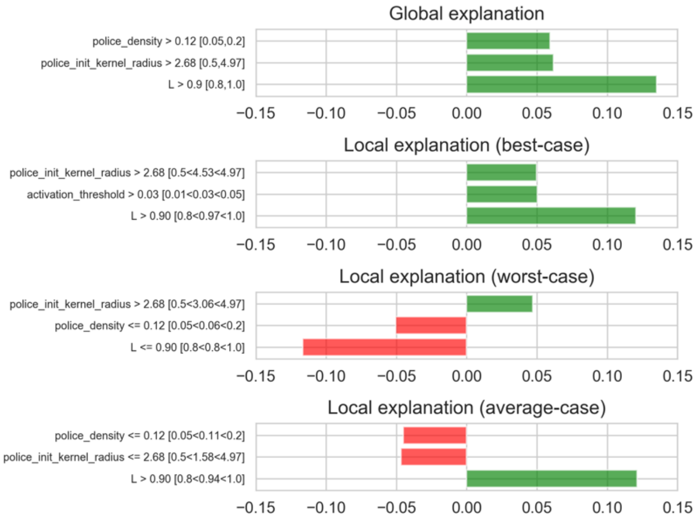


Figure 2. A LIME explainer on the output of the Epstein civil unrest model.

Epstein model (Figure 2). We begin with the global explanation. It suggests, as expected, that high legitimacy and police density are conducive to a lower peak in rioting (here represented as a numerically positive outcome). It also suggests that positioning police officers more uniformly, by increasing the kernel radius, has a similar effect – a uniform distribution of police forces is implied as best by this finding. A quick inspection of the remaining explanations shows that the role of these three variables remains consistent in all other explanations. The best-case example introduces an additional feature, the activation threshold (which determines the likelihood of rioting) as one of the most sensitive features. This suggests that in other areas of the space, the factors of police density, legitimacy and police spread are of most importance, but what sets apart the best-case example from other examples is that, additionally, it is very sensitive to the activation threshold. The "eruption" analogy that Epstein draws in his original paper comes to mind: the factors in the global explanation determine the "explosiveness" of the mix. For sufficiently explosive conditions, the activation threshold is perhaps less important, as the tipping point will inevitably be reached. However, for more stable situations (high legitimacy and effective policing), the likelihood of initial "sparks", which is controlled by the activation threshold, becomes a key factor.

Covid-19 model (Figure 3). We now turn our attention to the epidemiological model. Globally we observe that a high community transmission and infectiousness of mildly symptomatic individuals result in a higher number of deaths (here represented as a numerically negative outcome), while an increase in average duration of infectiousness has the opposite effect. This last result may look unintuitive, as one may reasonably expect that longer-lasting infectiousness would make the epidemic worse. Upon further inspection however, the correct interpretation of this parameter is more nuanced, and is related not just to the duration but also to the shape of the infectiousness profile[8]. In practice, decreasing this parameter results in a higher peak of infectiousness in the early stages of the infection followed by a steep drop, which can conceivably produce a net negative effect – this is indeed confirmed by the experiment. This is a good example of how explanations can help flag misinterpretations of the model parameters, which is a common danger with complex models.

Once again, the role of the three features in the global explanation remains consistent across different areas of the parameter space, but we observe a new feature entering the picture in the best-case scenario, where the infectiousness of asymptomatic individuals plays a more prominent role. This is perhaps related to the infectiousness profile being flatter here (because of a larger `mean_infectious_period`), increasing the impact of individuals that continue to sustain interactions throughout the infectious period due to lack of symptoms. This resonates with one of the key messages from the epidemiological community throughout the pandemic (e.g. Ghandi et al., 2020), that although this virus is neither the most contagious nor the most deadly we encountered in modern times, its ability to transmit asymptomatically places a limit to the extent it can be contained, even in optimistic forecasts.

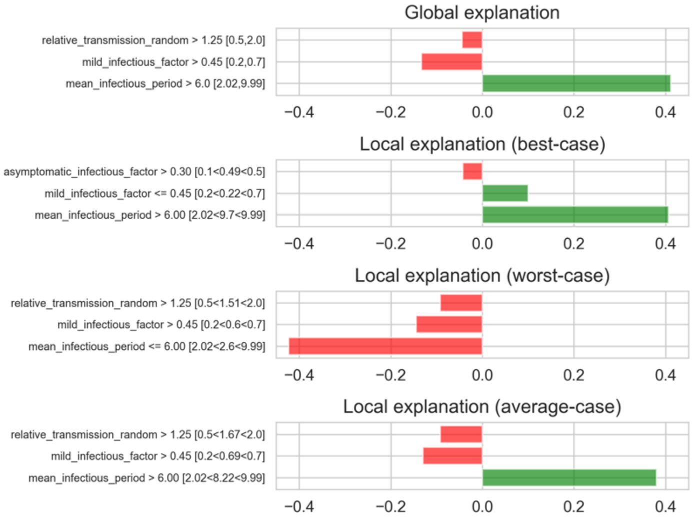


Figure 3. A LIME explainer on the output of the Covid-19 contact tracing model.

CONCLUSION

In this work we took first steps in the exploration of explainability methods as part of a toolkit of techniques, including sensitivity analysis, scenario analysis and calibration, aimed at making complex simulations more reliable, understandable and ultimately trustworthy. For LIME, the ability to "zoom in" on a specific simulation run is especially attractive, and makes for a natural complement to traditional scenario planning. Due to its generic formulation which treats the model as a black box, LIME is broadly applicable to a range of domain-specific models, and scales well to high dimensional parameter spaces via feature selection. We demonstrate how explanations can help build intuition on model behaviour without requiring a detailed understanding of the inner workings, and automatically surface discrepancies between user's expectations and what the model really does. We see this as a promising direction to enhance the utility of complex simulations in real-world decision making applications, where trust in the model plays a central role.

Several interesting directions for future research have been revealed by this work. For example, a powerful feature of LIME that we have not explored here is its ability to handle unstructured input such as image and text. The image format in particular is well-suited to represent the initial configuration of spatial ABMs, with agent coordinates and

[8] Namely, the infectiousness profile is modelled as a scaled gamma distribution with mean `mean_infectious_period` and a separate variance parameter.

internal states encoded as locations and colours on a map. This would enable the user to visually identify geographical areas that are especially important to the simulation dynamic ("hotspots"), taking full advantage of the spatial nature of the z_0 component of the model input. We reserve this investigation for a future work.

REFERENCES

Ahmad, M. A., Eckert, C., & Teredesai, A. (2018). Interpretable machine learning in healthcare. *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 559–560.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., & others. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. *International Conference on Machine Learning*, 173–182.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Muller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One*, *10*(7).

Boden, L. A., & McKendrick, I. J. (2017). Model-Based Policymaking: A Framework to Promote Ethical "Good Practice" in Mathematical Modeling for Public Health Policymaking. *Frontiers in Public Health*, *5*, 68.

Calder, M., Craig, C., Culley, D., de Cani, R., Donnelly, C. A., Douglas, R., Edmonds, B., Gascoigne, J., Gilbert, N., Hargrove, C., & others. (2018). Computational modelling for decision-making: Where, why, what, who and how. *Royal Society Open Science*, *5*(6), 172096.

Cannon, P., Greig, R., Boschi, G., & Anagnostopoulos, C. (2020). Machine learning surrogates for highly realistic simulations. *Proceedings of the The Interservice/Industry Training, Simulation and Education Conference (submitted)*.

Chermack, T. J., Lynham, S. A., & Ruona, W. E. (2001). A review of scenario planning literature. *Futures Research Quarterly*, *17*(2), 7–32.

Cioppa, T., Thomas, L., & Sanchez, S. (2004). Military applications of agent-based simulations. *Proceedings of the 2004 Winter Simulation Conference*, 171–180.

DeAngelis, D. L. (2018). *Individual-based models and approaches in ecology: Populations, communities and ecosystems*. CRC Press.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, *4*(5), 41–60.

Epstein, J. M. (2002). Modeling civil violence: An agent-based computational approach. *Proceedings of the National Academy of Sciences*, *99*(suppl 3), 7243–7250.

Epstein, J. M. (2006). *Generative social science: Studies in agent-based computational modeling*. Princeton University Press.

European Council. (2020). *Report on the comprehensive economic policy response to the COVID-19 pandemic.* (Press Release). Europa. https://www.consilium.europa.eu/en/press/press-releases/2020/04/09/report-on-the-comprehensive-economic-policy-response-to-the-covid-19-pandemic/

Ferretti, L., Wymant, C., Kendall, M., Zhao, L., Nurtay, A., Abeler-Dörner, L., Parker, M., Bonsall, D., & Fraser, C. (2020). Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, *368*(6491).

Fishman, G. S. (2013). *Discrete-event simulation: Modeling, programming, and analysis*. Springer Science & Business Media.

Ghandi, M., Yokoe, D., & Havlir, D. (2020). Asymptomatic transimission, the Achilles' heel of current strategies to control Covid-19. *The New England Journal of Medicine*.

Gilbert, N., & Terna, P. (2000). How to build and use agent-based models in social science. *Mind & Society*, *1*(1), 57–72.

Gipps, P. G., & Marksjö, B. (1985). A micro-simulation model for pedestrian flows. *Mathematics and Computers in Simulation*, *27*(2–3), 95–105.

Gu, W., Killeen, G. F., Mbogo, C. M., Regens, J. L., Githure, J. I., & Beier, J. C. (2003). An individual-based model of Plasmodium falciparum malaria transmission on the coast of Kenya. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, *97*(1), 43–50.

Hamby, D. (1995). A comparison of sensitivity analysis techniques. *Health Physics*, *68*(2), 195–204.

Hand, D. J. (2013). Data, not dogma: Big data, open data, and the opportunities ahead. *International Symposium on Intelligent Data Analysis*, 1–12.

HMG. (2015). *The Aqua Book*. https://www.gov.uk/government/publications/the-aqua-book-guidance-on-producing-quality-analysis-for-government

Hornberger, G. M., & Spear, R. C. (1981). Approach to the preliminary analysis of environmental systems. *J. Environ. Mgmt.*, *12*(1), 7–18.

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., & Aerts, H. J. (2018). Artificial intelligence in radiology. *Nature Reviews Cancer*, *18*(8), 500–510.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 4765–4774.

Murty, M. N., & Devi, V. S. (2011). *Pattern recognition: An algorithmic approach*. Springer Science & Business Media.

Nagel, K., & Rickert, M. (2001). Parallel implementation of the TRANSIMS micro-simulation. *Parallel Computing*, *27*(12), 1611–1639.

Pappenberger, F., Iorgulescu, I., & Beven, K. J. (2006). Sensitivity analysis based on regional splits and regression trees (SARS-RT). *Environmental Modelling & Software*, *21*(7), 976–990.

Ratto, M., Tarantola, S., & Saltelli, A. (2001). Sensitivity analysis in model calibration: GSA-GLUE approach. *Computer Physics Communications*, *136*(3), 212–224.

Ribeiro, M., Singh, S., & Guestrin, C. (2019). *Local Interpretable Model-Agnostic Explanations (LIME): An Introduction*. https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models* (Vol. 1). Wiley Online Library.

Taylor, L. J., Nabozny, M. J., Steffens, N. M., Tucholka, J. L., Brasel, K. J., Johnson, S. K., Zelenski, A., Rathouz, P. J., Zhao, Q., Kwekkeboom, K. L., & others. (2017). A framework to improve surgeon communication in high-stakes surgical decisions: Best case/worst case. *JAMA Surgery*, *152*(6), 531–538.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Windrum, P., Fagiolo, G., & Moneta, A. (2007). Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation*, *10*(2), 8.

World Health Organisation. (2020). *Covid-19 situation report* (No. 120; Coronavirus Disease 2019 (Covid-19): Situation Reports.). https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200519-covid-19-sitrep-120.pdf?sfvrsn=515cabfb_2