Analysis of COVID-19 Data with PRISM: Parameter Estimation and SIR Modelling

Paolo Milazzo(✉)

Department of Computer Science, University of Pisa,
Largo B. Pontecorvo, 3, 56127 Pisa, Italy
paolo.milazzo@unipi.it

Abstract. We propose a pipeline for the stochastic analysis of a SIR model for COVID-19 through the stochastic model checker PRISM. The pipeline consists in: (i) the definition of a modified SIR model, able to include governmental restriction and prevention measures through an additional time-dependent coefficient; (ii) parameter estimation based on real epidemic data; (iii) translation of the modified SIR model into a Continuous Time Markov Chain (CTMC) expressed using the PRISM input language; and (iv) stochastic analysis (simulation and model checking) with PRISM.

Keywords: PRISM model checker · SIR models · COVID-19

1 Introduction

The impact that the COVID-19 (or better, SARS-COV-2) pandemic is having on the population around the world is recorded in increasingly large and varied databases. The spread of the virus is tracked on a daily basis almost everywhere in the world, but the effects of the epidemics can be observed also in datasets in the contexts of healthcare, mobility, finance, and many others. The analysis of COVID-19 epidemic data could help in understanding the dynamics of the contagion, evaluating the effect of restriction and prevention measures taken by national and local governments and predicting the effect of alternative measures.

Epidemic phenomena are often studied by means of a *SIR model* [8]. This happened also for COVID-19 pandemic, with several extensions of the model proposed to take into account its peculiarities [2,5–8]. SIR models typically describe epidemics as *deterministic* dynamical systems, through Ordinary Differential Equations (ODEs). For a more realistic description of the epidemic dynamics, *stochastic fluctuations* are often to be taken into account. This happens, in particular, when a *small number* of infected individuals are present in the population, causing the disease spread to depend tightly on the probability of such few individuals to meet and infect other people. In order to deal with these stochastic events, SIR models can be reformulated in terms of *Continuous Time Markov Chains (CTMCs)*. This can be done essentially by interpreting

J. Bowles et al. (Eds.): DataMod 2020, LNCS 12611, pp. 123–133, 2021.
https://doi.org/10.1007/978-3-030-70650-0_8

infection and recovery rates, already used in ODEs, as parameters of exponential distributions. The obtained CTMC can then be analyzed through suitable methods which include, for example, stochastic simulation.

PRISM [1] is one of the most used probabilistic/stochastic model checkers. It can be used to study dynamical properties of a CTMC through an *exhaustive exploration* of all possible behaviors. Dynamical properties can be expressed as temporal logic formulae (for CTMCs, PRISM supports the CSL temporal logic [3]). Properties assessment consists then in an exhaustive exploration of the CTMC state space This may require a long sequence of matrix multiplications giving the probability distribution of each possible state at discrete time steps.

Stochastic model checking allows studying properties of a dynamical systems in a very systematic way. Property assessment does not provide only information about the possible systems behaviors: given a dynamical property (e.g. reachability of a given state, causality between events or possibility of oscillation), a stochastic model checker computes the probability that the system behavior will satisfy it. This analysis is not performed on a bunch of simulation results, but by taking all possible behaviors into account. Of course, the main limitation of stochastic model checking techniques is often due to size of the state space (state explosion problem). Moreover, in the case of stiff systems, property assessment may require a huge number of matrix multiplications. In some cases, these limitations can be overcome by using suitable model specification tricks.

In this paper, we describe preliminary processing and modelling activities that allow a SIR model of the COVID-19 pandemic to be analyzed with PRISM. Our approach actually consists in the following pipeline:

1. Definition of a *modified SIR model* (based on ODEs) that allows taking into account restriction and prevention measures (e.g. lockdown);
2. Parameter estimation using standard Python libraries (NumPy and SciPy);
3. Translation of model into a CTMC expressed in the PRISM input language;
4. Analysis with PRISM.

We will use real data about the spread of COVID-19 in the Tuscany Region (Italy) to show the pipeline steps. However, our aim is not to perform a deep analysis of such data with PRISM, but to show how it is possible to obtain, from data, a PRISM model that can be analyzed efficiently. Hence, although we will show some inferences and analysis results, the intended contribution of this paper is mostly methodological.

2 SIR Epidemic Models and COVID-19

Epidemic phenomena are often studied by means of a *SIR model* [8]. The SIR acronym summarizes the classes of individuals into which the population is partitioned. They are: *Susceptible*, individuals who can be infected; *Infected*, individuals who have been infected and that can infect susceptible ones; and *Recovered*, individuals who passed the infection phase and can no longer infect others.

The dynamics of epidemic phenomena is described by means of a system of Ordinary Differential Equations (ODEs). In its simplest formulation, the

model includes one equation for each class of individuals. The population size is assumed constant over time and it is *normalized* in $[0, 1] \subseteq \mathbb{R}$. Hence, variables S, I, $R \in [0, 1]$ with $S + I + R = 1$ describe the *ratios* of each class of individual in the population. Moreover, the model is based on the following assumptions:

- infection and recovery are the only relevant events: other events related to reproduction, death, migration, etc., are not taken into account;
- disease is transmitted by personal contacts between individuals of I and S classes (*horizontal transmission*);
- contacts between individuals are random, i.e. the number of infections is proportional to both I and S;
- after infection and recovery, individuals become resistant to the disease.

Therefore, the model is described by this small system of differential equations:

$$\begin{cases} \frac{dS}{dt} = -\beta SI \\ \frac{dI}{dt} = \beta SI - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases} \tag{1}$$

where β is the *infection coefficient*, describing the probability of infection after the contact of a healthy individual with an infected one, and γ is the *recovery coefficient*, describing the rate of recovery of each infected individual (in other words, $1/\gamma$ is the time one individual requires for recovering). Note that:

- S can only decrease, and R can only increase;
- if $\beta < \gamma$ (i.e., $\beta/\gamma < 1$), I can only decrease (since $S \leq 1$);
- if $\beta > \gamma$ (i.e., $\beta/\gamma > 1$), the behavior of I depends on S. It initially increases if $S > \gamma/\beta$.

Many extensions of the SIR model are available in the literature, and have been proposed to study different infection schemes, the effects of vaccinations or the influence of information. In order to apply the SIR model to the COVID-19 epidemic and, in particular, in order to analyze data collected during the first few months of the epidemic, it is necessary to take into account prevention measures (e.g. lockdown) that have been enforced by the national governments. Hence, we propose a variant of the SIR model which includes a time dependent coefficient $p(t)$ expressing the effect of such measures on the infection rate.

Our *modified SIR model* is hence defined as follows:

$$\begin{cases} \frac{dS}{dt} = -\beta SIp(t) \\ \frac{dI}{dt} = \beta SIp(t) - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases} \tag{2}$$

where $p(t) \in [0, 1] \subset \mathbb{R}$ is used to scale down the infection coefficient β in accordance with the strength of the enforced prevention measures at time t. A value of $p(t)$ close to 0 represents strong prevention, while $p(t) = 1$ means no prevention at all. Let us consider the first few weeks of the epidemics, and let us assume that lockdown has been enforced at time t_{lock}. With some degree of approximation, we can describe $p(t)$ as a piecewise linear function as follows:

$$p(t) = \begin{cases} 1 & \text{if } t < t_{lock} \\ p_{lock} & \text{if } t \geq t_{lock} \end{cases} \tag{3}$$

with $p_{lock} \in [0, 1] \subset \mathbb{R}$ modeling the effect of lockdown on infection coefficient.

3 Parameter Estimation

Now, we face the problem of estimating parameters for the modified SIR model presented in (2). In particular, if we assume $p(t)$ to be expressed by a piecewise linear function as in (3), we have to estimate values for β, γ, and p_{lock}.

By focusing on the Tuscany Region, we can estimate such parameters by applying standard optimization methods in order to fit real epidemic data. We use COVID data published on a daily basis by the Regional Health Agency of the Tuscany Region[1]. The dataset[2] includes data on infections, deaths, hospitalizations, etc., collected every day in the whole region starting from February 24th, 2020. Moreover, data on infections are available also disaggregated by province.

We focus on the time period of March-May 2020, corresponding to the initial spread of the infection and the lockdown phase. More precisely, we consider the time interval between day 20 (March, 15th) and day 75 (May, 9th). We choose not to consider data from the first 20 days since the number of detected infections in that period is extremely small, and probably unreliable.

In order to take into account geographical distribution of the population in the Tuscany Region, we choose to use the (modified) SIR model at the level of provinces. This choice will mitigate the assumption of the SIR model that the population is uniformly distributed in the territory, and that all individuals can freely meet with each other. Moreover, this will allow us to evaluate and compare differences in the disease spread in different provinces.

Tuscany consists of ten provinces. Some of them (e.g. Prato and Firenze) have a high population density, while others (e.g. Grosseto and Siena) are large and less populated. Since population density could have a correlation with the infection rate, considering data at the level of provinces could lead to more accurate parameter estimations.

The Python scripts we developed for parameter estimation purposes are available as a Jupyter Notebook on GitHub[3]. In order to estimate the parameters of the SIR model for the different provinces, we use functionalities provided by standard Python packages. In particular, we use the `optimize.curve_fit` function of the SciPy library, to find optimal values for coefficients β, γ and p_{lock}.

We apply `curve_fit` twice: the first time to estimate β and γ on the basis of the pre-lockdown data (hence, by assuming $p(t) = 1$)), and the second time to estimate p_{lock} by assuming β and γ as estimated before and by using lockdown data. As value for t_{lock} in (3), namely, as time for the enforcement of lockdown measures, we choose 45, namely April 9th. Actually, in Italy the lockdown state

[1] Agenzia Regionale di Sanita (ARS), https://www.ars.toscana.it/.
[2] Freely available at http://dati.toscana.it/dataset/open-data-covid19.
[3] GitHub repository: https://github.com/Unipisa/SIR-covid.

has been reached through a sequence of governmental measures taken in the period between March 5th (schools closed) and March 22nd (national lockdown). The effects of such measures on the epidemic dynamics started to become evident more than two weeks later, hence around April, 9th.

Let us assume a Python function `ModelSolution(t,beta,gamma,prev,x0)` that uses the `odeint` solver provided by the SciPy package to solve ODEs of the modified SIR model in (2), with `t` a sequence of time point for which to solve the ODEs, `beta` and `gamma` corresponding to β and γ, respectively, `prev` a constant value for $p(t)$, and `x0` an array of initial conditions (i.e., initial values for S, I and R). We define function `f1` and we pass it to `curve_fit` as follows:

```
f1 = lambda t,beta,gamma: ModelSolution(t,beta,gamma,1,x0)
p1 = curve_fit(f,t1,pre_lockdown_data,bounds=(0,[np.inf,1]))
```

The result `p1` contains two optimal values for β and γ with $\beta \in [0, \infty)$ and $\gamma \in [0, 1]$, that fit pre-lockdown data.

Now, we define function `f2` and we pass it to `curve_fit` as follows:

```
f2 = lambda t,prev: ModelSolution(t,p1[0][0],p1[0][1],prev,x0)
p2 = curve_fit(f,t2,lockdown_data,bounds=(0,1))
```

Result `p2` contains now an optimal value for p_{lock}, fitting lockdown data.

For both optimizations, it is important to point out our choice for the initial condition array `x0`. More precisely, it is important to clearly explain how we relate variables S, I and R with real data. The number of infected individuals reported in the dataset is the number of persons that resulted positive to a SARS-COV-2 test. After the test, these persons are then isolated and have a very small probability of infecting other people. So, individuals reported as infected in the dataset have a role in the epidemic that is actually more similar to that of a recovered individual than of an infected one. The "real" infected individuals are instead those that have been infected, but have not been identified yet through a specific SARS-COV-2 test. These behave mostly as healthy individuals and infect other people. Unfortunately, these "real" infected individuals are hidden in the population and their number is unknown. In the initial array `x0` for the first optimization step, we choose to set the initial value of I as the triple of the number of positive persons reported on March, 15th. This because we assume that in the initial phases of the epidemic only a small part of the positive individuals were identified. The condition array `x0` for the second optimization step simply correspond to the final state reached after the first optimization.

Parameters resulted from the described estimation process are reported in Table 1. Apart form the Arezzo province, whose estimated parameters look

Table 1. Parameters estimation.

	β	γ	p_{lock}
AREZZO	0.229187	0.251815	0.994549
FIRENZE	0.145179	0.097259	0.001654
GROSSETO	0.129687	0.144080	0.487087
LIVORNO	0.107479	0.104104	0.317674
LUCCA	0.120928	0.111307	0.004195

	β	γ	p_{lock}
MASSA CARRARA	0.102454	0.084304	0.000098
PISA	0.122128	0.127283	0.472081
PRATO	0.130999	0.119076	0.145995
PISTOIA	0.078007	0.099515	0.991426
SIENA	0.077028	0.069914	0.000231

like outliers, all provinces exhibit an infection coefficient β in the interval $[0.077, 0.145]$ and a recovery coefficient γ in $[0.06, 0.127]$. Provinces with a high population density, such as Firenze and Prato, actually correspond to highest infection coefficients. The estimation of p_{lock} is instead less regular, thus suggesting that something could be improved about the modelling of the lockdown effect. Inaccuracies could also be caused by the low quality of measurements in the first period of the pandemic. Anyway, the estimated p_{lock} values provide useful qualitative information about the areas in which lockdown has given better results.

Figures 1 and 2 show numerical simulation results of the modified SIR model (only the curves of I and R are depicted) compared with the real data about cumulative number of infected individuals (dots). The curve of I is actually a *prediction*, since, as we already explained, we use I to represent "real" infected individuals that are hidden in the population. The shape of this curve, that in many cases shows an edge at the start of lockdown, demonstrates the positive effect of such a prevention measure.

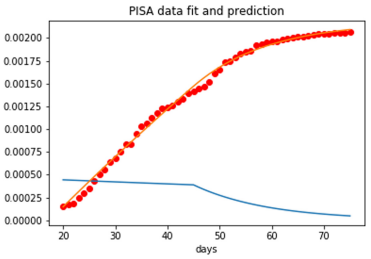

Fig. 1. Data fitting and predictions (Pisa province)

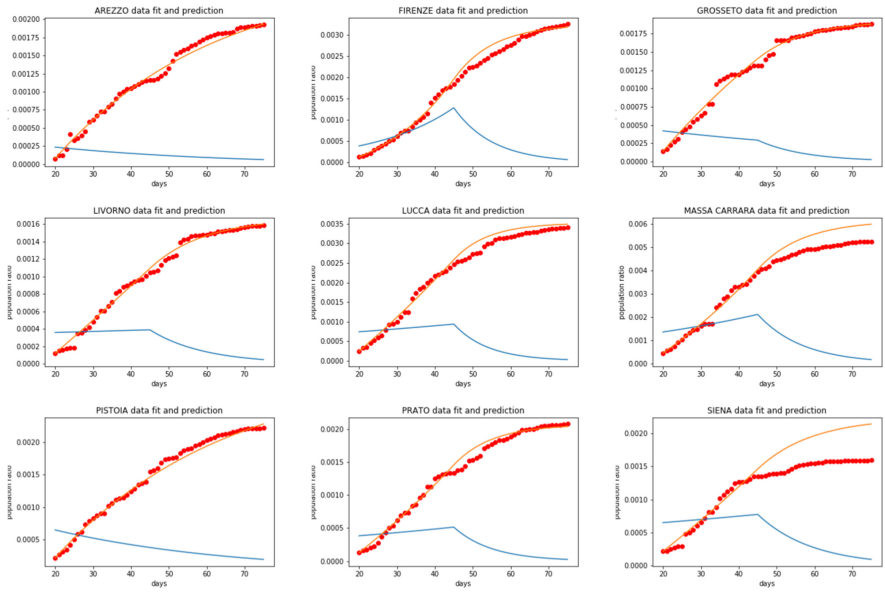

Fig. 2. Data fitting and predictions (other provinces)

4 Translation into CTMC and Analysis with PRISM

The next step we perform is to translate our extended SIR model into a stochastic model, by discretizing variables and by considering infection and recovery rates as parameters of a Continuous Time Markov Chain (CTMC). This allows us to obtain a model that is, in principle, more accurate in capturing the epidemic dynamics, by taking into account random fluctuations that may have a significant role in the case of small numbers of infected individuals.

Dynamical properties of the obtained CTMC could then be analyzed using the stochastic model checker PRISM [1,9]. Stochastic model checking, compared for instance to analysis by stochastic simulation, allows computing in a systematic way the probability of occurrence of emerging behaviors with specific properties of interest. The main problem of model checking is, however, its poor scalability to models with a very large state space. A stochastic SIR model representing a population of hundreds of thousand of individuals (like in a Tuscan province) can be very likely affected from this kind of scalability problems.

A way to solve scalability issues can be to resort to *statistical* model checking methods: a variant of stochastic model checking which provides approximate results by exploiting stochastic simulation result. PRISM itself has built-in statistical model checking facilities. However, before considering this solution, there are a few modelling tricks that can significantly reduce the state space.

PRISM describes CTMC states through a set of *bounded integer variables*. Since ODEs of the SIR model are based on real variables, the first step we have to perform is to *discretize* the model. Hence, we assume a discretization constant `SIZE` and we replace the variables domain $[0.0, 1.0] \subset \mathbb{R}$ with $[0..SIZE] \subset \mathbb{N}$.

This leads to the following naive CTMC specification in PRISM input language, where model parameters are defined by the `beta`, `gamma` and `plock` constants (initialized with estimations for the province of Pisa), `SIZE` is the discretization constant, `s`, `i` and `r` are the model variables (again, initialized with values from data collected on the province of Pisa) and we have two transitions describing events of infection and recovery, respectively.

```
ctmc

const double beta = 0.122128; const double gamma = 0.127283;
const double plock = 0.472081; const int SIZE = 100000;

module SIR_Pisa

s : [0..SIZE] init 99936;
i : [0..SIZE] init 48;
r : [0..SIZE] init 16;

[] i>0 & i<SIZE & s>0 -> beta*s*i*plock/SIZE : (s'=s-1)&(i'=i+1);
[] i>0 & r<SIZE -> gamma*i*plock : (i'=i-1)&(r'=r+1);

endmodule
```

The problem of this translation is that, by assuming `SIZE = 100000`, the state space turns out to include 10^{15} *potentially reachable states*, which make the model computation and analysis by PRISM unfeasible.

A first refinement of the model can be obtained by observing that one of the three variables `s`, `i` and `r` can be pruned. Indeed, as in the original ODEs we had $S + I + R = 1$, in the PRISM counterpart we always have `s + i + r = SIZE`. Removing, for instance, `s` will require to make a small change to the definition of the first transition, where `s` has to be replaced by `SIZE-(i+r)`.

Pruning variable `s` immediately reduces the state space, bringing it to a size of 10^{10} states. However, this is still too huge for PRISM.

As a second refinement, we choose to introduce an upper bound to the number of infected and of recovered individuals. For example, we choose these numbers to be always smaller than 500. As shown in the following CTMC specification, where also the first refinement is implemented, this can be obtained by adding a new constant `BOUND` that is then used to define the domain of the two variables `i` and `r`. Moreover, we have to explicitly change the model transition to describe the behavior in the case the upper bound is reached. The two transitions of the naive translation have to be enabled only when `i` and `r` are strictly smaller than `BOUND`. Moreover, it is necessary to introduce a third transition that, in case the number of recovered individuals reaches the upper bound, allows an infected individual to recover (i.e. it decreases `i` by one) without increasing `r`.

```
ctmc

const double beta = 0.122128; const double gamma = 0.127283;
const double plock = 0.472081; const int SIZE = 100000; const int BOUND = 500;

module SIR_Pisa

i : [0..BOUND] init 48;
r : [0..BOUND] init 16;

[] i>0 & i<BOUND -> beta*(SIZE-(i+r))*i*plock/SIZE : (i'=i+1);
[] i>0 & r<BOUND -> gamma*i*plock : (i'=i-1)&(r'=r+1);
[] i>0 & r=BOUND -> gamma*i*plock : (i'=i-1);

endmodule
```

The addition of the upper bound actually makes the model approximated. However, if the upper bound is high enough to make the probability of the variables to reach it negligible, we have that the approximation will have no influence on the probabilities of dynamical properties assessed through model checking. We remark that the assumption on the small number of infected individuals was one of the motivations for the use of a stochastic modelling approach. In the case of big numbers, that could lead to unfeasible models with large state spaces, the whole stochastic approach would be poorly motivated, since with big numbers stochastic fluctuations would become much less relevant.

Upper bounds significantly reduce the state space, that now turns out to include "only" 250000 *states*. This makes model construction and analysis with PRISM very fast, in particular (and this is *very important*) if either the *sparse* or the *explicit* engines are selected in the relevant PRISM settings menu.

As examples of analyses performed with PRISM, we show in Fig. 3 some results of stochastic simulation and model checking performed using parameters of the Pisa province and by comparing lockdown and no-lockdown scenarios.

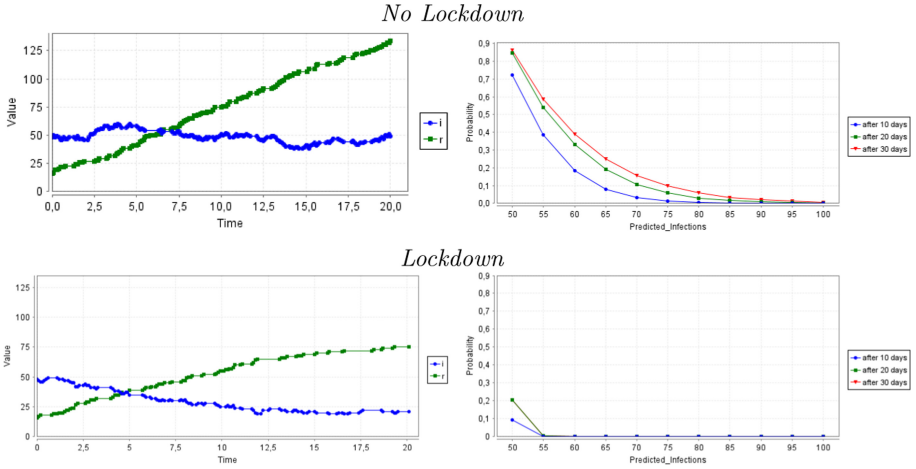

Fig. 3. Analysis of the Pisa model with PRISM (examples). On the left, a single run of a stochastic simulation. On the right, probabilities, computed by stochastic model checking, of reaching given numbers of infected individuals after 10, 20 and 30 days. The CSL property used for model checking is `P=? [F<XX i=Predicted_Infections]`, where `XX` is 10, 20 or 30, and `Predicted_Infections` takes values as in the graph.

Simulations show that lockdown can effectively reduce the number of infected individuals, leading to a slow down of the disease spread. As before, this model allows understanding the dynamics of hidden infected individuals. Model checking is used to make predictions on the future number of infected individuals, by computing probabilities of reaching different threshold values in 10, 20, 30 days.

Stochastic model checking can be used to make predictions about reachable population states in an accurate, systematic and efficient way. This makes this technique a good candidate for real time epidemic monitoring and decision support. Moreover, the modified SIR model could be extended to describe also, for instance, age classes, hospitalizations, new therapies or vaccinations. In this case, it would be possible to use stochastic model checking as a tool to evaluate hypotheses about these new aspects, for instance by computing the probability of disease eradication when alternative vaccination strategies are followed.

5 Conclusions

In this paper we proposed a *pipeline* for the stochastic analysis of a SIR model for COVID-19 through the stochastic model checker PRISM. The whole pipeline is informative: in the parameter estimation phase, the estimated parameters themselves provide useful information about the different dynamics in different areas (e.g., provinces) and about the effectiveness of restriction and prevention measures such as lockdown. Moreover, by performing numerical simulation of the deterministic models used for parameter estimations we were able to predict the dynamics of hidden positive individuals.

PRISM allows defining a stochastic SIR model in *a dozen lines of code*. An optimized model can be analyzed in a *few minutes*. The analysis performed through the model checking features of PRISM is *exhaustive*, and not based only on a few simulation runs. These positive performance results have been obtained by applying a couple of modelling tricks (variable pruning and upper bounds) that allowed state space of the model constructed by PRISM to be reduced by several orders of magnitude. The introduction of upper bounds to the values of variables actually introduces a small approximation in the model, that is negligible in practically relevant cases. As a consequence, we believe that this approach aimed at making the analysis with PRISM feasible is in this case preferable to approaches based, for instance, on statistical model checking techniques. Indeed, the latter techniques would base the model checking analysis on stochastic simulation results, losing exhaustivity.

This paper aimed at proposing the modelling and analysis methodology. Developments of the approach could include improving the modelling of the restriction measures by considering more accurate definitions of the $p(t)$ function in the modified SIR model. Function $p(t)$ could be defined in order to gradually change after the enforcement of prevention measures, or in order to depend on the current infection trend (if the number of infected individual increases, people tends to be more cautious). Moreover, extensions of the model including age classes, hospitalizations, new therapies or vaccinations could be defined. Further work would include performing a deeper analysis of COVID data with PRISM, also by taking some of these additional aspects into account, even when some parameters about these aspects are not precisely known [4].

Acknowledgements. This work is supported by the Università di Pisa under the "PRA – Progetti di Ricerca di Ateneo" (Institutional Research Grants) - Project no. PRA_2020-2021_26 "Metodi Informatici Integrati per la Biomedica".

References

1. PRISM Probabilistic Model Checker. https://www.prismmodelchecker.org/
2. Acemoglu, D., et al.: A multi-risk SIR model with optimally targeted lockdown. Tech. rep., National Bureau of Economic Research (2020)
3. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Verifying continuous time Markov chains. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 269–276. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61474-5_75
4. Barbuti, R., Levi, F., Milazzo, P., Scatena, G.: Probabilistic model checking of biological systems with uncertain kinetic rates. Theor. Comput. Sci. **419**, 2–16 (2012)
5. Calafiore, G.C., Novara, C., Possieri, C.: A modified SIR model for the COVID-19 contagion in Italy. arXiv preprint arXiv:2003.14391 (2020)
6. Chen, Y.C., Lu, P.E., Chang, C.S.: A time-dependent SIR model for COVID-19. arXiv preprint arXiv:2003.00122 (2020)
7. D'Arienzo, M., Coniglio, A.: Assessment of the SARS-CoV-2 basic reproduction number, R0, based on the early phase of COVID-19 outbreak in Italy. Biosaf. Health **2**, 57–59 (2020)

8. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. R. Soc. Lond. Ser. A Contain. Pap. Math. Phys. Character **115**(772), 700–721 (1927)
9. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47