Covid-19 infection in Italy. Mathematical models and predictions

A comparison of logistic and exponential models applied to Covid-19 virus infection in Italy.

Gianluca Malato Follow
Mar 8 · 6 min read ★

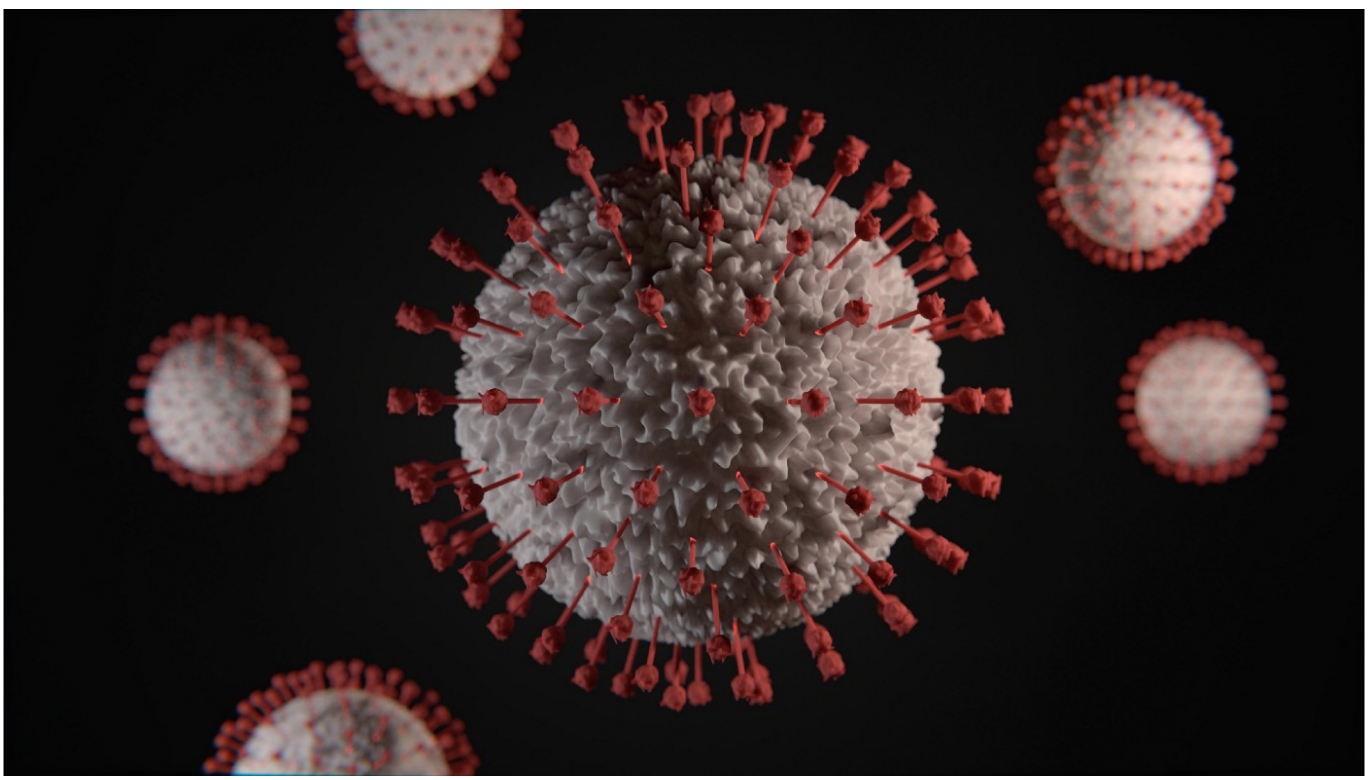


Photo by Viktor Forgacs on Unsplash

The world is fighting against a new enemy in these days, which is the **Covid-19** virus.

The virus has spread quickly in the world since its first appearance in China. Unfortunately, **Italy** is recording the **highest number** of Covid-19 infected people **in Europe**. We've been the **first nation** facing this new enemy in the Western World and we are all fighting every day against all the **economical and social** implications of this virus.

In this article, I'll show you a simple **mathematical** analysis of the infection growth in Python and **two models** to better understand the evolution of the infection.

. . .

Data collection

Every day, the Italian Civil Protection Department refreshes the cumulative data of infected people. This data is **publicly available** as open data on GitHub here: https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv

My goal is to create **models** of the time series of the **total number of infected people to date** (i.e. the actually infected people plus the people who have had been infected). These models have **parameters**, which will be estimated by **curve fitting**.

Let's do it in Python.

First, let's import some libraries.

```
import pandas as pd
import numpy as np
from datetime import datetime,timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
import matplotlib.pyplot as plt
%matplotlib inline
```

Now, let's take a look at the **raw data**.

```
url = "https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv"

df = pd.read_csv(url)
```

	data	stato	ospedalizzati	isolamento_domiciliare	attualmente_positivi	dimessi_guariti	deceduti	totale_casi	nuovi_attualmente_positivi
0	2020-02-24 18:00:00	ITA	127	94	221	1	7	229	221
1	2020-02-25 18:00:00	ITA	149	162	311	1	10	322	90
2	2020-02-26 18:00:00	ITA	164	221	385	3	12	400	74
3	2020-02-27 18:00:00	ITA	304	284	588	45	17	650	203
4	2020-02-28 18:00:00	ITA	409	412	821	46	21	888	233
5	2020-02-29 18:00:00	ITA	506	543	1049	50	29	1128	228
6	2020-03-01 18:00:00	ITA	779	798	1577	83	34	1694	528
7	2020-03-02 18:00:00	ITA	908	927	1835	149	52	2036	258
8	2020-03-03 18:00:00	ITA	1263	1000	2263	160	79	2502	428
9	2020-03-04 18:00:00	ITA	1641	1065	2706	276	107	3089	443
10	2020-03-05 18:00:00	ITA	2141	1155	3296	414	148	3858	590
11	2020-03-06 18:00:00	ITA	2856	1060	3916	523	197	4636	620
12	2020-03-07 18:00:00	ITA	3218	1843	5061	589	233	5883	1145

The column we need is 'totale_casi' which contains the cumulative number of infected people to date.

This is the raw data everything starts from. Now, let's **prepare** it for our analysis.

Data preparation

First, we need to change dates into **numbers**. We'll take the days since January 1st.

```
df = df.loc[:,['data','totale_casi']]

FMT = '%Y-%m-%d %H:%M:%S'

date = df['data']

df['data'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01 00:00:00", FMT)).days  )
```

	data	totale_casi
0	54	229

1	55	322
2	56	400
3	57	650
4	58	888
5	59	1128
6	60	1694
7	61	2036
8	62	2502
9	63	3089
10	64	3858
11	65	4636
12	66	5883

We can now analyze the two models I'll take into the exam, which are the **logistic function** and the **exponential function**.

Each model has **three parameters**, that will be estimated by a **curve fitting** calculation on the historical data.

The logistic model

The logistic model has been widely used to describe the **growth of a population.** An infection can be described as the growth of the population of a pathogen agent, so a logistic model seems **reasonable**.

This formula is **very known** among data scientists because it's used in the logistic regression classifier and as an activation function of neural networks.

The most generic expression of a logistic function is:

$$f(x,a,b,c) = \frac{c}{1+e^{-(x-b)/a}}$$

In this formula, we have the variable *x* that is the time and three parameters: *a,b,c*.

- *a* refers to the infection speed
- *b* is the day with the maximum infections occurred
- *c* is the total number of recorded infected people at the infection's end

At high time values, the number of infected people **gets closer and closer** to *c* and that's the point at which we can say that the infection **has ended**. This function has also an **inflection point** at *b*, that is the point at which the first derivative **starts to decrease** (i.e. the peak after which the infection starts to become less aggressive and decreases).

Let's define it in python.

```
def logistic_model(x,a,b,c):
    return c/(1+np.exp(-(x-b)/a))
```

We can use the *curve_fit* function of *scipy* library to estimate the parameter values and errors starting from the original data.

```
x = list(df.iloc[:,0])
y = list(df.iloc[:,1])

fit = curve_fit(logistic_model,x,y,p0=[2,100,20000])
```

Here are the values:

- *a*: 3.54
- *b*: 68.00
- *c*: 15968.38

The function returns the **covariance matrix** too, whose diagonal values are the variances of the parameters. Taking their square root we can calculate the standard errors.

```
errors = [np.sqrt(fit[1][i][i]) for i in [0,1,2]]
```

- Standard error of a: 0.24
- Standard error of b: 1.53
- Standard error of c: 4174.69

These numbers give us many useful insights.

The **expected number of infected people** at infection end is 15968+/- 4174.

The **infection peak** is expected around 9 March 2020.

The **expected infection end** can be calculated as that particular day at which the cumulative infected people count **is equal** to the c parameter rounded to the nearest integer.

We can use the *fsolve* function of *scipy* to numerically find the root of the equation that defines the infection end day.

```
sol = int(fsolve(lambda x : logistic_model(x,a,b,c) - int(c),b))
```

It's on 15 April 2020.

Exponential model

While the logistic model describes ain infection growth that is **going to stop** in the future, The exponential model describes an **unstoppable** infection growth. For example, if a patient infects 2 patients per day, after 1 day we'll have 2 infections, 4 after 2 days, 8 after 3 and so on.

The most generic exponential function is:

$$f(x,a,b,c) = a \cdot e^{b(x-c)}$$

The variable *x* is the time and we still have the parameters *a, b, c*. The meaning, however, is different from the logistic function parameters'.

Let's define the function in Python and let's perform the same curve fitting procedure used for logistic growth.

```
def exponential_model(x,a,b,c):
    return a*np.exp(b*(x-c))

exp_fit = curve_fit(exponential_model,x,y,p0=[1,1,1])
```

Parameters and their standard errors are:

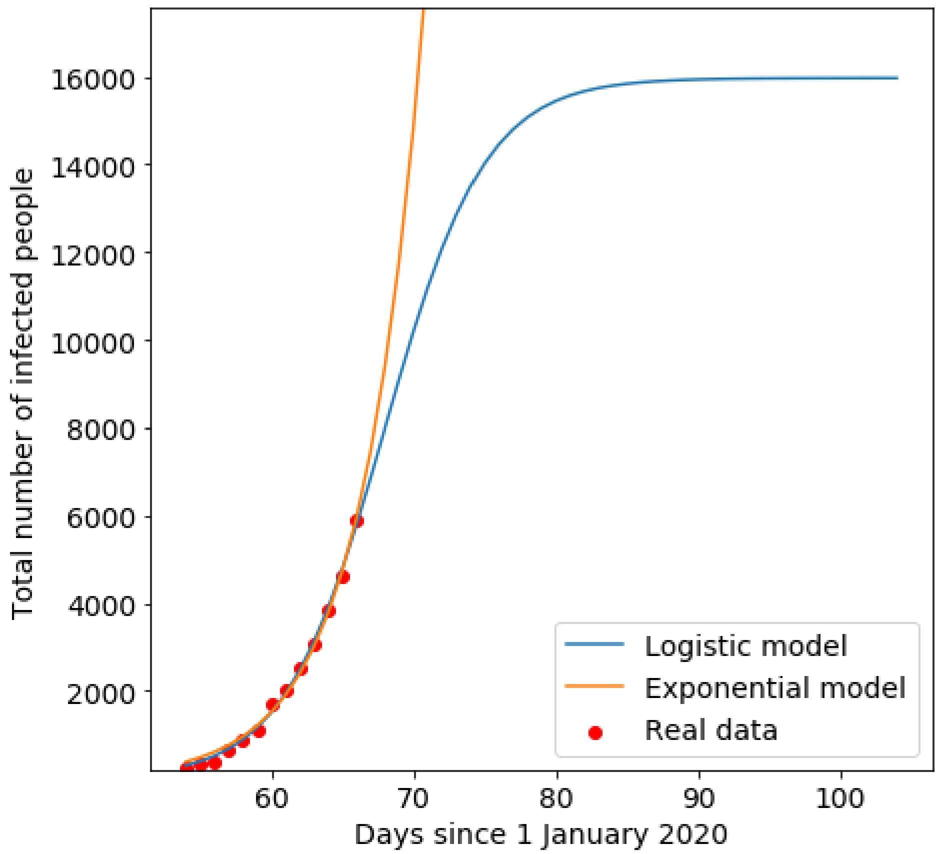
- *a*: 0.0019 +/- 64.6796
- *b*: 0.2278 +/- 0.0073
- *c*: 0.50 +/- 144254.77

Plots

We have now all the necessary data to visualize our results.

```
pred_x = list(range(max(x),sol))
plt.rcParams['figure.figsize'] = [7, 7]

plt.rc('font', size=14)

# Real data
plt.scatter(x,y,label="Real data",color="red")

# Predicted logistic curve
plt.plot(x+pred_x, [logistic_model(i,fit[0][0],fit[0][1],fit[0][2])
for i in x+pred_x], label="Logistic model" )

# Predicted exponential curve
plt.plot(x+pred_x, [exponential_model(i,exp_fit[0][0],exp_fit[0]
[1],exp_fit[0][2]) for i in x+pred_x], label="Exponential model" )

plt.legend()
plt.xlabel("Days since 1 January 2020")
```

```
plt.ylabel("Total number of infected people")
plt.ylim((min(y)*0.9,c*1.1))

plt.show()
```



Both theoretical curves seem to approximate the experimental trend quite well. Which one does it better? Let's take a look at the **residuals**.

Analysis of residuals

Residuals are the **differences** between each experimental point and the corresponding theoretical point. We can analyze the residuals of both models in order to verify the **best fitting curve**. In a first approximation, the lower **Mean Squared Error** between theoretical and experimental data, the **better** the fit.

```
y_pred_logistic = [logistic_model(i,fit[0][0],fit[0][1],fit[0][2])
for i in x]

y_pred_exp =  [exponential_model(i,exp_fit[0][0], exp_fit[0][1],
exp_fit[0][2]) for i in x]
```

```
mean_squared_error(y,y_pred_logistic)
mean_squared_error(y,y_pred_exp)
```

Logistic model MSE: 8254.07

Exponential model MSE: 16219.82

Which is the right model?

Residuals analysis seems to point toward the **logistic model**. It's very likely because the **infection should end** someday in the future; even if everybody will be infected, they'll develop the proper **immunity defense** to avoid a second infection. That's right as long as the virus **doesn't mutate** too much (as, for example, influenza virus).

But there's something that **still worries me**. I've been fitting the logistic curve every day since the beginning of the infection and every day **I got different parameter values**. The number of infected people at the end **increases**, the maximum infection day is often the current day or the next day (which is compatible with the standard error of 1 day on this parameter). That's why I think that, although the logistic model seems to be the most reasonable one, the shape of the curve **will probably change** due to exogenous effects like new infection **hotspots**, government **actions to bind** the infection and so on.

That's why I think that the predictions of this model will start to become useful only within a few weeks, reasonably after the infection peak.

. . .

Note from the editors: towardsdatascience.com is a Medium publication primarily based on the study of data science and machine learning. We aren't health professionals or epidemiologists. To learn more about the coronavirus pandemic, you can click here.

- Coronavirus does not spare the young.
- What a nuclear submarine captain knows about self-isolating.
- The truth about vitamin D, zinc, and other coronavirus rumors.
- ’I’d rather be here’ — an expat perspective from South Korea.

Data Science Immunology Epidemiology Coronavirus