

Using Sector Based Linear Regression in Epidemic Data

G. Nagy*

*Óbuda University, Alba Regia Technical Faculty, Institute of Geoinformatics

nagy.gabor@amk.uni-obuda.hu

ORCID ID: 0000-0001-9453-2291

Abstract—The statistical analysis and the machine learning are important tools of the research nowadays. In the last one and half year the importance of epidemic data and the analysis of this data has been increased as a result of the COVID-19 pandemic. This paper describes a research which uses Sector Based Linear Regression (SBLR), a robust linear regression for the analyzing and predicting with epidemic data.

Index Terms—robust regression, SBLR, Sector Based Linear Regression, COVID-19, epidemic data

I. INTRODUCTION

The analysis of the collected data was very important at all times, but especially nowadays, when the computers can make highly difficult calculations for the better result. The international databases provide large amount of data about the COVID-19 epidemic, and we would like to discover the correlations and predict the numbers of following days.

A. The analysis and prediction of epidemic data

The data of COVID-19 epidemic is published by many different authorities and organizations in the countries of the World. The data relates countries or lower regions spatial, and typically daily in time. The data collecting and integrating is a hard work, but fortunately there are repositories with the unified data. I use the data repository of the Centre for System Science and Engineering (CSSE) at John Hopkins University [1], [2].

This dataset can be downloaded from GitHub¹. The dataset contains many CSV (coma separated values) files with different values (confirmed cases, deaths, recovered, etc) by date and country (or lower level region). The CSV is a simple and well usable format for the programs, which will be introduced in the remaining part of this article.

B. The SBLR method

The Sector Based Linear Regression (SBLR) is an robust linear regression method [3], this is a generalized case of the Fitting Disc Method [4], which was developed for point cloud processing. The principle of SBLR is shown in two dimension in the Figure 1. The method looks for a line so, that this line divides the control points to two equal set in each sector (the right and the left side of the area).

The SBLR may be used with different quantiles, but in this paper I use only $q = 0.5$.

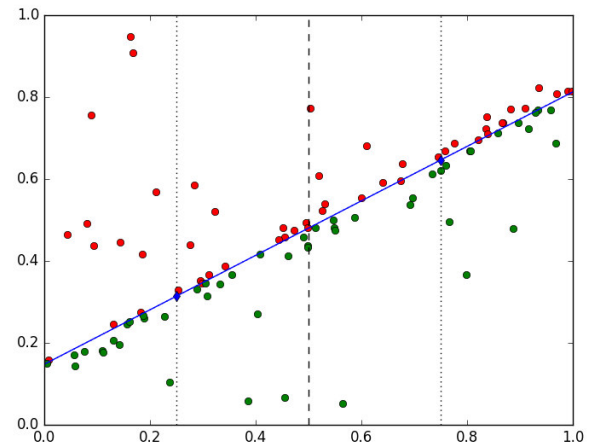


Figure 1. The principle of SBLR method. The half of the control points are above, and the another half are under the regression line in each sector. This is an robust regression, because the outlier control points do not modify significantly the regression line.

II. LINEAR AND EXPONENTIAL REGRESSION

The epidemic waves have a phase when the growth is exponential. The prediction of the values is very important in this days.

The SBLR provides an linear regression method. The method can be applicable, if we use the logarithms of the values. The predicted values (for the following days) will be calculated by the exponent function.

This modification is suitable, if the dataset does not contain 0 values. In the analysis of this paper I selected time intervals without 0 data.

III. DATA AND SOFTWARE

There is a Python [5], [6], [7] module, which has been implemented the SBLR method:

```
"""
Module for Sector Based Linear Regression
(See in: DOI: 10.14232/actacyb.23.4.2018.3)
Author: Nagy Gábor
"""

import numpy

def regnd(n):
    """The coordinates of the nodes of a regular n-dimensional hypertetrahedron,
    which centre is in the origin, and the distances
    of the nodes from the origin are 1"""
    if n==0:
```

¹<https://github.com/CSSEGISandData/COVID-19>

```

    return [[]]
else:
    return [ [ y*(1-(1/n)**2)**0.5 for y in x]+[-1/n]\
              for x in regnd(n-1)]+[[0.0]*(n-1)+[1.0]]

def scaledl(lst, lm):
    """Scale a list of coordinates by lm"""
    return [ [ y*lm for y in x] for x in lst]

def lintr(lst, la, lb):
    """Calculate the linear transformation of the coordinates
    of the lst by x'=ax+b with the parameters of la and lb."""
    return [ [ la[i]+li[i]+lb[i] for i in range(len(lst[0]))] for li in lst]

def calctr(ptlst):
    """Calculate the transformation parameters for linear transform
    to (-1,...,-1)-(+1,...,+1) n-dimension cube every point of the lst"""
    nr=range(len(ptlst[0]))
    if len(ptlst)<2:
        return ([l for i in nr], [-ptlst[0][i] for i in nr])
    mincoord=lst(map(min, numpy.transpose(ptlst)))
    maxcoord=lst(map(max, numpy.transpose(ptlst)))
    apar=[ 2/(maxcoord[i]-mincoord[i]) for i in nr]
    return (apar, [-1-mincoord[i]*apar[i] for i in nr])

def minidx(lst):
    """Return the index of the minimal value of the list."""
    wlst=lst(lst)
    ret=0
    for i in range(1,len(wlst)):
        if wlst[i]<wlst[ret]:
            ret=i
    return ret

def linpar2cth(linpars):
    """Calculate centre heights from the linear parameters."""
    dim=len(linpars)-1
    ctcoords=[x+[l] for x in scaledl(regnd(dim), dim/(dim+1))]
    return [sum([x[i]*linpars[i] for i in range(dim+1)]) for x in ctcoords]

def cth2linpar(cths, chidx=-1, chcoord=[]):
    """Calculate linear parameters from the centre height."""
    dim=len(cths)-1
    ctcoords=[x+[l] for x in scaledl(regnd(dim), dim/(dim+1))]
    wcths=lst(cths)
    if chidx>-1:
        ctcoords[chidx]=chcoord[:dim+1]
        wcths[chidx]=chcoord[-1]
    return lst(numpy.linalg.solve(ctcoords, wcths))

def sbllr(ptlst, q=0.5):
    """Calculate linear equation to the pointlist by SBLR method"""
    dim=len(ptlst[0])-1
    ctps=scaledl(regnd(dim), dim/(dim+1))
    ctcoords=[x+[l] for x in ctps]
    ptsep=[[] for x in range(dim+1)]
    pa,pb=calctr(ptlst)
    pa[-1]=1
    pb[-1]=0
    for pti in lintr(ptlst, pa, pb):
        ptsep[minidx([sum([(pti[j]-ctps[i][j])**2 for j in range(dim)])\
                          for i in range(dim+1)])].append(pti)
    if min(map(len, ptsep))<1:
        raise ValueError('Wrong distribution, empty sector!')
    cths=numpy.percentile([x[-1] for x in si], q*100 for si in ptsep)
    fullstep=0
    noch=0
    secid=0
    while noch<dim+1 and fullstep<100*dim+2:
        oldcth=cths[secid]
        cths[secid]=numpy.percentile([linpar2cth(cth2linpar(cths, secid, x))\
                                     for x in ptsep[secid]], q*100)
        if abs(oldcth-cths[secid])>1E-5:
            noch=0
        else:
            noch+=1
            secid=(secid+1)%dim+1
            fullstep+=1
    retpar=cth2linpar(cths)
    return [retpar[i]*pa[i] for i in range(dim)]+\
           [sum([retpar[i]*pb[i] for i in range(dim)])+retpar[-1]]

```

This module can be downloaded from GitHub².

The SBLR module provides linear regression in an n -dimensional space, but I use only the two dimensional case in this paper.

IV. TESTING THE METHOD

I use 28 days intervals from the exponential growth phases of the waves. This is four full week, the dataset contains four-point from every day of the week.

The data series, the regressed curve are shown in Matplotlib [8], [9] diagrams.

The Figure 2 shows the result of a regression. The orange line is an exponential curve, which has been fit to the last 28 days values marked by blue. The predicted curve is the red line, this is the continuation of the fitted line. Because this

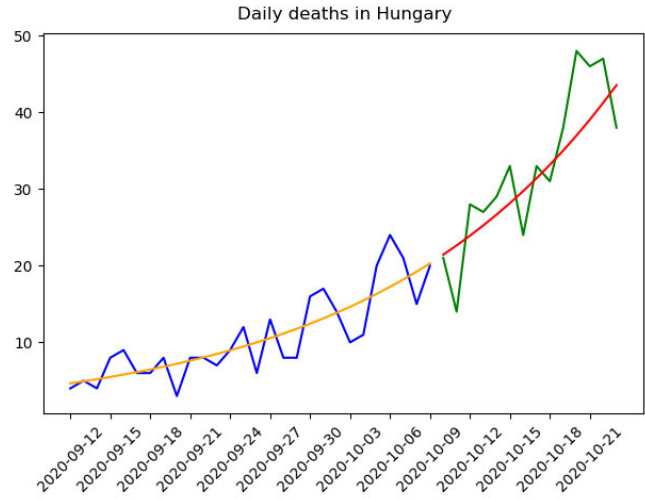


Figure 2. The predicted deaths in Hungary in the beginning of the second wave.

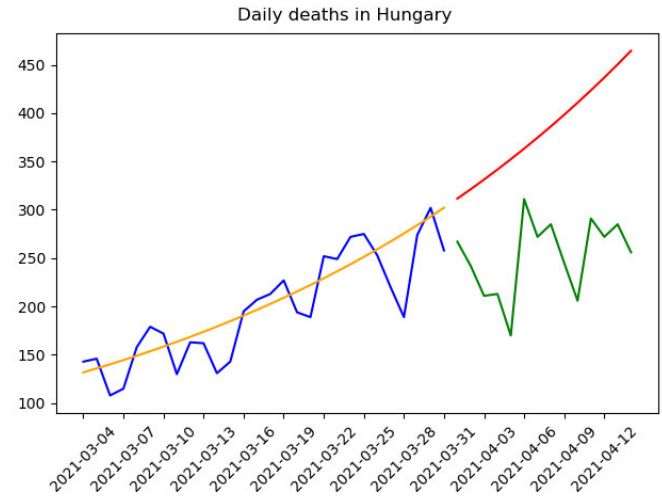


Figure 3. The predicted deaths in Hungary in the top of the third wave. The prediction is wrong, because the growth had stopped.

data is from the last year, I can draw the realized values by the green line.

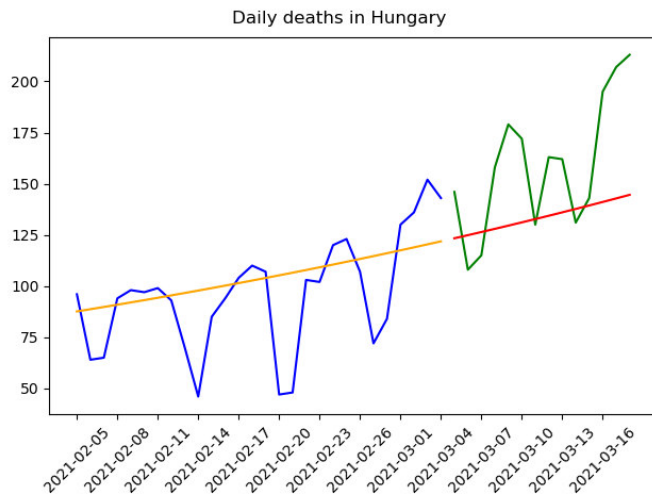
The described method can predict only the exponential phase of a wave. The Figure 3 presents an unwell regression in the top of the third wave, when the growth was stopped.

V. CONCLUSION

The SBLR method is an robust linear regression method. This method may be applied with exponentially increasing case, if we use the logarithms of the values.

The described method can predict the values of the following days from the data of the previous period, but the predictions are correct only the exponential growth phase.

²<https://github.com/ngabor/sblr>



[9] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, May-Jun 2007.

Figure 4. The predicted deaths in Hungary in the beginning of the third wave.

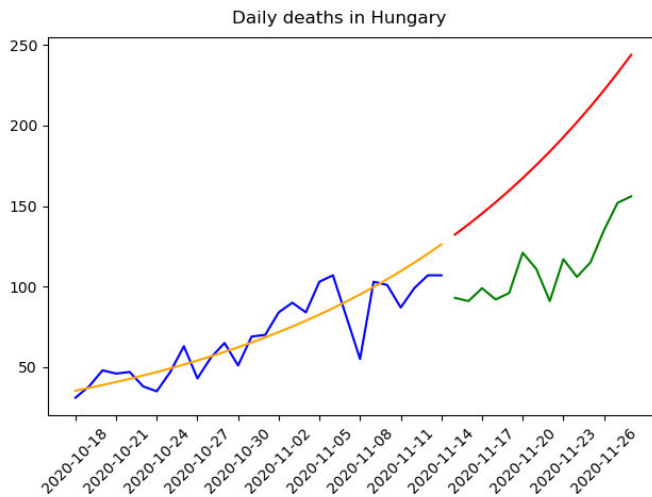


Figure 5. An unwell regression from the growing phase of the second wave.

REFERENCES

- [1] J. CSSE, "Covid-19 data repository by the center for systems science and engineering (csse) at johns hopkins university," 2020. [Online]. Available: <https://github.com/CSSEGISandData/COVID-19>
- [2] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet infectious diseases*, vol. 20, no. 5, pp. 533–534, 2020.
- [3] G. Nagy, "Sector based linear regression, a new robust method for the multiple linear regression," *ACTA CYBERNETICA*, vol. 23, no. 4, pp. 1017–1038, 2018.
- [4] G. Nagy, T. Jancsó, and C. Chen, "The fitting disc method, a new robust algorithm of the point cloud processing," *ACTA POLYTECHNICA HUNGARICA*, vol. 14, no. 6, pp. 59–73, 2017.
- [5] G. Van Rossum *et al.*, "Python Programming Language." in *USENIX Annual Technical Conference*, vol. 41, 2007.
- [6] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, 2007.
- [7] K. J. Millman and M. Aivazis, "Python for scientists and engineers," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 9–12, 2011.
- [8] G. Nagy, "Matplotlib in the geospatial analyzes," in *12th International Symposium on Applied Informatics and Related Areas (AIS 2017)*, G. T. Orosz, Ed. Óbudai Egyetem, 2017, pp. 90–93.