

Exercise: COVID modeling

The question considered in this problem is: can you predict the number of new cases of COVID each day in a county of a state based on the number of new cases in the other counties of the state? To answer this, suppose there are m counties in the state, and the respective number of new COVID cases are x_1, x_2, \dots, x_m . If x_m refers to the county of interest, then the assumption (i.e., the model function) is

$$x_m = v_1x_1 + v_2x_2 + \dots + v_{m-1}x_{m-1}.$$

a) Suppose the values of the x_j 's are measured every day for n days. Let x_{ij} be the number of new COVID cases for county x_j on the i th day. The error function is still given in (8.15), but now \mathbf{A} is $n \times (m - 1)$ and \mathbf{y} is a n -vector. What are \mathbf{A} and \mathbf{y} in this problem in terms of the x_{ij} 's?

You need two data matrices \mathbf{T} (the training data) and \mathbf{S} (the testing data). How to obtain these matrices is explained on the next page, or you can use the data files provided (which are for New York state with x_m corresponding to Rensselaer County).

b) Once \mathbf{T} and \mathbf{S} are known, then $\mathbf{A}=\mathbf{T}(:,1:m-1)$; and $\mathbf{y}=\mathbf{T}(:,m)$; . Compute \mathbf{v} using the Moore-Penrose pseudo-inverse.

c) For the testing data set, the predicted values for x_m are: $\mathbf{yp}=\mathbf{S}(:,1:m-1)\mathbf{v}$; . Also, the measured values for x_m are: $\mathbf{ym}=\mathbf{S}(:,m)$; . Plot \mathbf{yp} versus \mathbf{ym} (this should be a scatter plot). Include in this plot the 45° line (if the \mathbf{yp} 's are correct they should lie on this line).

d) Redo parts (b) and (c) but use the QR approach to compute \mathbf{v} .

e) The agreement between the measured values and the predicted values is really poor, really good, or something in-between. Which is it? Try to explain this result. For example, is it realistic to expect that you can predict what is happening on any given day in county x_m using what is happening in other places on that same day across the state? Keep in mind that the counties in the state vary significantly (some very rural, some densely populated, some with different rules about quarantining, etc).

f) Comment on any differences between the predictions using the Moore-Penrose pseudo-inverse and the QR approach.

g) Redo the exercise but redo the randomized splitting of the data (as described on the next page). How much does the answer depend on the particular splitting?

Steps to obtain the data matrices **T** and **S**

In the COVID data spreadsheet, remove everything except the data for the counties for your state from 5/1/20 onwards. You are to then read the data into MATLAB, and then transform the data into the format assumed in the model. A step by step approach to this is below.

Step 1: Either use the included spreadsheet (which was downloaded on 12/2/22) or download the current data for CASES from the website:

<https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/>

The rows in this spreadsheet list the counties for each state and the columns give the daily cumulative number of COVID cases for the respective county (starting with 1/22/20). Delete the columns for all dates from 1/22/20 to 4/30/20 (so the first date is now 5/1/20). After this remove all rows except those for the counties of your state. Pick your county and let its row number be i_c . With this, then remove the first 3 columns so the first column now corresponds to the cases on 5/1/20. Save the resulting spreadsheet as a “tab delimited text” file. In what follows it’s assumed that this file is named data.txt.

Step 2: Read in the data file into MATLAB using the command: `D=importdata('data.txt');`. The columns of **D** are the daily cumulative number of cases in each county. You need **N**, which is the matrix giving the number of new cases. If \mathbf{n}_j is the j th column of **N**, and \mathbf{d}_j is the j th column of **D**, then $\mathbf{n}_j = \mathbf{d}_{j+1} - \mathbf{d}_j$ (so, **N** has one less column than **D**). In MATLAB this can be accomplished using the command: `N(:,j)=D(:,j+1)-D(:,j);`. After determining **N**, take its transpose, and then interchange column i_c with column m (so now the data for your county is in the last column). Call the resulting matrix **X**.

Step 3: Split the rows of **X** into a training set **T**, which consists of a random selection of approximately 2/3 of the rows from **X**, and a testing set **S** which are the other 1/3. In MATLAB, this can be done as follows:

```
rows=randperm(n);
nn=round(2*n/3);
T=X(rows(1:nn),:);
S=X(rows(nn+1:n),:);
```

Step 4: Center the data based on the daily average for the training data. The MATLAB commands for this are:

```
meanT=mean(T);
nT=length(T(:,1));
for j=1:m
    T(:,j)=T(:,j)-meanT(j)*ones(nT,1);
    S(:,j)=S(:,j)-meanT(j)*ones(n-nT,1);
end
```