Regional Research Institute
West Virginia University

Technical Document Series

MATLAB and Python Functions for Weighted Disaggregation

Amir Borges Ferreira Neto and Randall W. Jackson

RRI TechDoc 2015-02

Date submitted: 07/31/2015
Key words/Codes: Matlab, Python, Weight Disaggregation; C67, R13

MATLAB and Python Functions for Weighted Disaggregation*

Abstract

This technical document describes a weighted disaggregation problem and provides the code for a function in MATLAB and Python to perform such disaggregation.

Problem Context

Finding data in the desired level of disaggregation is not always possible. A common practice to deal with such problem is to disaggregate the data following some scheme defined by the researcher, and use some reference data to create a weight vector that will be used to calculate the data available in the desired level of disaggregation.

Example 1:

There is a country with a production structure comprising 10 industries. The information available is as follows:

- Output reported for four sectors
- The correspondence between the four sectors and the 10 disaggregate industries, and
- Employment for each of the 10 industries

*Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. 1235684 and USDA NIFA Award 2012-67009-19660.

With the assumption that output per employee is the same for all industries within the sector, the industrial employment distribution can be used to disaggregate the sector output data (See figure 1). Let N be the number of industries and K the number of sectors. Define:

1. **D** (K x 1) is the vector of data to be disaggregated
2. **A** (N x 1) is the vector with the aggregation scheme defined by the researcher
3. **R** (N x 1) is the vector of reference data used to disaggregate vector **D**, and
4. **O** (N x 1) is the vector with the output, i.e., the disaggregated data of interest.

Hence to calculate **O**:

$$\mathbf{M} = \mathbf{S} * \mathbf{R} \tag{1}$$

$$\mathbf{T} = \mathbf{S'} * \mathbf{M} \tag{2}$$

$$\mathbf{W} = (\hat{\mathbf{T}})^{-1} * \mathbf{R} \tag{3}$$

$$\mathbf{Z} = \mathbf{S'} * \mathbf{D} \tag{4}$$

$$\mathbf{O} = \mathbf{W} \circ \mathbf{Z} \tag{5}$$

where, **S** (K x N) is an aggregation matrix , **M** (K x 1) is an auxiliary matrix with the sum of the reference data (**R**) for each sector, **T** (N x 1) is vector where the totals calculated in **M** (eq. (1)) are assigned to the respective industries, **W** (N x 1) is the weight vector that is going to be used to disaggregate the data in vector **D**, **Z**(N x 1) is an auxiliary matrix assigning the data to be disaggregated (sector level) to its respective industries, and **O** is the output matrix with the disaggregated data per industry. The $\hat{\mathbf{T}}$ represents a diagonalization operation in vector **T**.

Back to the example, **A** is the aggregation scheme mapping the 10 industries to 4 sectors, **R** is the employment data for the 10 industries and **D** is the output for the 4 sectors. Calculating the industry output (**O**), using

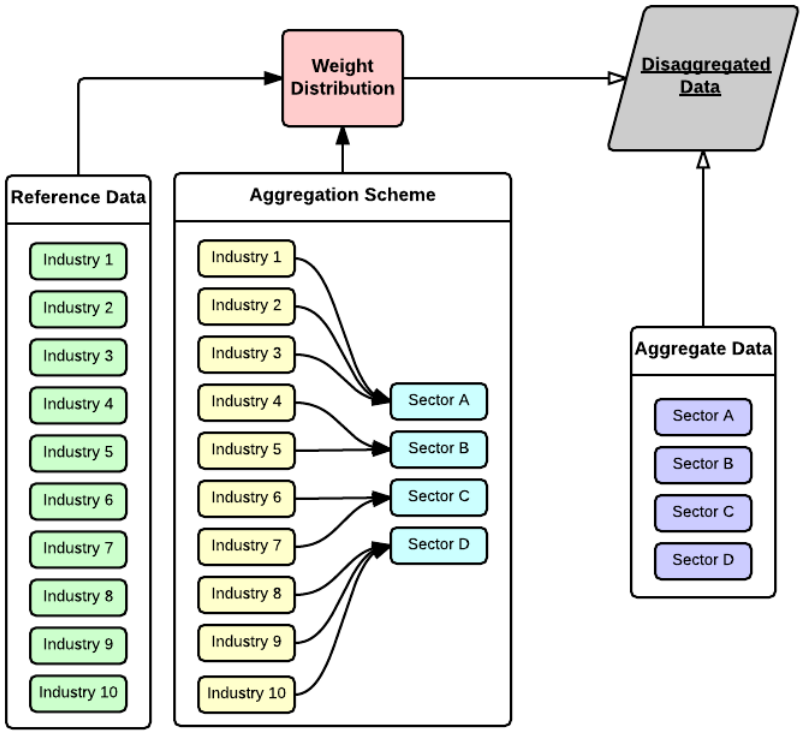

Figure 1 - Calculation process for disaggregated data

the aggregation scheme of Figure 1, and letting $\mathbf{D'} = \begin{bmatrix}10 & 20 & 30 & 40\end{bmatrix}$ and $\mathbf{R'} = \begin{bmatrix}10 & 20 & 15 & 15 & 20 & 10 & 10 & 5 & 10 & 15\end{bmatrix}$ the results are as follows (see Figure 1):

$$\mathbf{S} = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1\end{bmatrix}; \mathbf{M} = \begin{bmatrix}45\\35\\20\\30\end{bmatrix}; \mathbf{T} = \begin{bmatrix}45\\45\\45\\35\\35\\20\\20\\30\\30\\30\end{bmatrix};$$

$$\mathbf{W} = \begin{bmatrix}0.222\\0.444\\0.333\\0.428\\0.572\\0.500\\0.500\\0.166\\0.333\\0.500\end{bmatrix}; \mathbf{K} = \begin{bmatrix}10\\10\\10\\20\\20\\30\\30\\40\\40\\40\end{bmatrix}; \mathbf{O} = \begin{bmatrix}2.222\\4.444\\3.333\\8.571\\11.429\\15.000\\15.000\\6.666\\13.333\\20.000\end{bmatrix}$$

Supporting Algorithm(s)/Code

Matlab 1

```
function [O]=wdisag(A,R,D)
% PURPOSE: Computes a disaggregated vector using weights from a defined
% variable
%
% This program has been developed to disaggregate information taking into
% account some weight structure defined by some aggregation scheme and
% some reference variable.
% Example:
% Disaggregate GDP taking into account the employment structure of sectors
%
```

```
% USAGE:
% A = Aggregation scheme
% R = Reference Data to construct weight structure
% D = Aggregated data
%
% RESULT:
% O = Disaggregated vector taking into account weight structure
%
%-------------------------------------------------------------------------
% This code was developed by Amir B. Ferreira Neto and Randall W. Jackson
%-------------------------------------------------------------------------


n = max(D);       % Finds the number of aggregated data (e.g. sectors)
Total = zeros(1,n);  % Creates the vector of aggregated data to use as weight
W = zeros(1,length(R));  % Creates the vector with weight values
DD = zeros(1,length(R)); % Creates the disaggregated vector
for i=1:length(R)
    T(A(i)) = Total(A(i))+R(i); % Accumulation of each value of Total
end


for i=1:length(R)
    W(i) = R(i)./T(A(i)); % Calculate the weights
end


for i=1:length(R)
    O(i) = W(i).*D(A(i));  % Calculate the disaggregated values
end

csvwrite(Disagg_vector, O)
```

Matlab 2

```
function [O]=wdisag2(A,R,D)
% PURPOSE: Computes a disaggregated vector using weights from a defined
% variable
%
% This program has been developed to disaggregate information taking into
% account some weight structure defined by some aggregation scheme and
% some reference variable.
% Example:
% Disaggregate GDP taking into account the employment structure of sectors
```

```
%
% USAGE:
% A = Aggregation scheme (N x 1)
% R = Reference Data to construct weight structure (N x 1)
% D = Aggregated data (K x 1)
%
% where N is the number of industries and K the number of sectors
%
% RESULT:
% O = Disaggregated vector taking into account weight structure (N x 1)
%
% ____________________
% When running the function the following auxiliary vectors will be created
% S = Aggregagation scheme (please refer to RRI Tech Doc 2013-04) (K x N)
% M = Sum of reference data per sector (K x 1)
% T = Assignment of data from M (sector level) to corresponding industries
% W = Weight vector to disaggregate data
% Z = Auxiliary matrix assigning aggregate data to corresponding industries
%
%-------------------------------------------------------------------------
% This code was developed by Amir B. Ferreira Neto and Randall W. Jackson
%-------------------------------------------------------------------------

% ____________________
% This part of the code was developed by Caleb Stair and you can download
% it at www.rri.wvu.edu. This function is presented as RRI Tech Doc 2013-04
% The author has granted permission to embed his code in this function.

lr=length(A);
S=sparse(max(A),lr);
for i=1:lr
   S(A(i),i)=1;
end
% ____________________

M = S*R;
T = S'*M;
W = inv(diag(T))*R;
Z = S'*D;
O = W.*K;

csvwrite(Disagg_vector, O)
```

Python

```
# PURPOSE: Computes a disaggregated vector using weights from a defined variable
#
# This program has been developed to disaggregate information taking into"
# account some weight structure defined by some aggregation scheme and
# some reference variable.
# Example:
# Disaggregate GDP taking into account the employment structure of sectors
#
# USAGE:
# A = Aggregation scheme
# R = Reference Data to construct weight structure
# D = Aggregated data
#
# RESULT:
# O = Disaggregated vector taking into account weight structure
#
#_____________________________________________________________________________
# This code was developed by Amir B. Ferreira Neto and Randall W. Jackson
#_____________________________________________________________________________

def wdisag(A, R, D):
    import numpy as np
    n = max(A) # Finds the number of aggregated data (e.g. sectors)
    T =np.zeros(n) # Creates the vector of aggregated data to use as weight
    W = np.zeros(len(R)) # Creates the vector with weight values
    O = np.zeros(len(R))  # Creates the disaggregated vector

    for i in range(len(RD)):
        T [A[i]-1] += R[i]  # Accumulation of each value of Total

    for i in range(len(RD)):
        W[i] = R[i]/T[A[i]-1] # Calculate the weights

    for i in range(len(RD)):
        O[i] = W[i]*D[A[i]-1]  # Calculate the disaggregated values

    import csv
    O_out = open('Wdisag.csv', 'wb')
    Owriter = csv.writer(O_out, dialect='excel')
    Owriter.writerow(O)
    O_out.close()
```