Instructions to run the dynamic solar adoption diffusion model

1 Brief Description

This tool predicts the number of solar rooftop adopters over time in rural Virginia. The model runs on two regions - Rappahannock county and Shenandoah Valley Region (SVR). SVR is a fairly large area as compared to Rappahannock county. Hence, the peer networks and other associated data for SVR are large. The peer networks for SVR are stored in the database which the end user will have to download on their machine. Rappahannock data can be efficiently read and processed via text files because of their small size. Household demographics and neighborhood information is used to understand the diffusion of solar adopters in these regions. A regression model is built to determine household level probability of adoption and neighborhood features are dynamically updated at every time step. This document provides steps to use this model.

The procedure outlined below works with Windows and Linux operating systems.

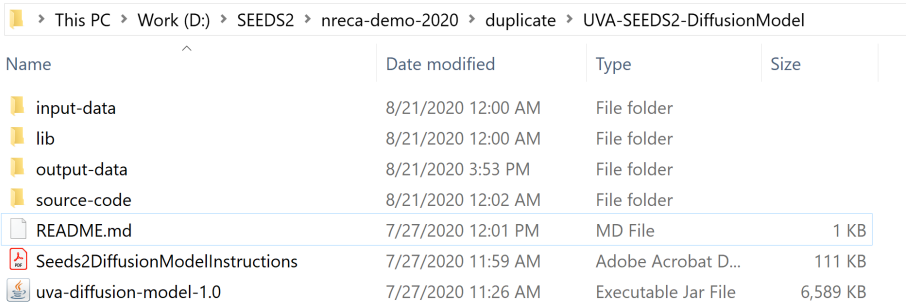

Figure 1: Sample base directory structure of the project on a Windows system

2 Pre-requisites

2.1 Software requirements

1. Java 1.8
   Download link: `https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html`
   Instructions: `https://java.com/en/download/help/download_options.xml`

2. Sqlite3
   Download link : `https://www.sqlite.org/download.html`
   Instructions: `http://www.sqlitetutorial.net/download-install-sqlite`

3. Git
   Download link: `https://git-scm.com/downloads`

2.2 Project setup

1. Navigate to the desired directory on your computer to clone/download the project. Download the github repository using the following command:
   git clone https://github.com/NSSAC/UVA_SEEDS2.git

   The repository provides two models – static and dynamic. Please navigate to the 'dynamic-diffusion-model' directory.

   Note: *baseDir* = Full folder path upto the project directory *UVA_SEEDS2/dynamic-diffusion-model*
   e.g.
   On Windows: `D:\\SEEDS2\\gitCode\\UVA_SEEDS2\\dynamic-diffusion-model`
   On Linux/Unix: `/project/seeds2/UVA_SEEDS2/dynamic-diffusion-model`

2. Unzip the following files under the folder *baseDir*/input-data/svr :
   (a) svr-hh-records.zip
   (b) svr-mapping.zip

3. Unzip the following files under the folder *baseDir*/input-data/rappahannock :
   (a) rappahannockMiles1.zip
   (b) rappahannockMiles3.zip
   (c) rappahannockMiles4.zip

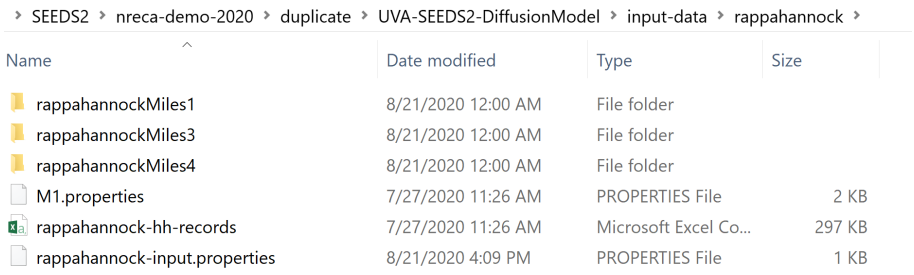

Figure 2: Rappahannock - The input directory structure after unzipping required folders.

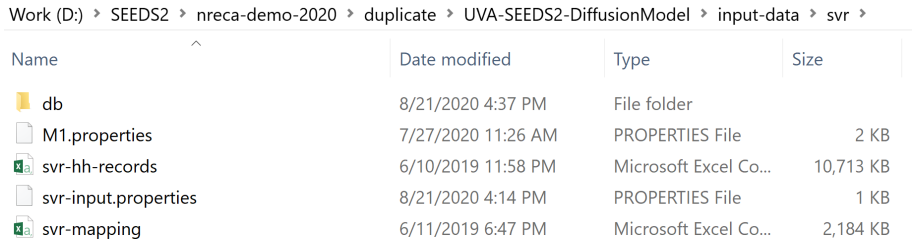

Figure 3: SVR - The input directory structure after unzipping required folders.

4. Copy the household neighborhood network database file from `https://dataverse.lib.virginia.edu/dataset.xhtml?persistentId=doi:10.18130/V3/HDTHSN` to the *baseDir*/input-data/svr/db folder.

5. As examples two input files are provided for running the tool on 2 regions (SVR and Rappahannock). These input files are named as svr-input.properties and rappahannock-input.properties. Update the path of the base directory in rappahannock-input.properties and svr-input.properties files. This will be the 'baseDir' parameter in these files. These files are present under the *input-data* directory:

   baseDir/input-data/

   [Note: .properties files can be opened in any text editor.
   If one forward slash / does not work for Windows machine, then, try two backward slashes \\ in the path for the variable *baseDir* in svr-input.properties and rappahannock-input.properties]

3 Running the model

To execute the diffusion model, the user will require to execute the file named *uva-diffusion-model-1.0.jar* . Navigate to the *baseDir* directory.
To run the model for Rappahannock region:
On Windows operating system execute the following command:
java -cp *baseDir*/lib/sqlite-jdbc-3.23.1.jar;uva-diffusion-model-1.0.jar uva.nssac.model.DiffusionModeller *baseDir*/input-data/rappahannock/rappahannock-input.properties
On Linux/Unix operating system execute the following command:
java -cp *baseDir*/lib/sqlite-jdbc-3.23.1.jar:uva-diffusion-model-1.0.jar uva.nssac.model.DiffusionModeller *baseDir*/input-data/rappahannock/rappahannock-input.properties

To run the model for SVR region:
On Windows operating system execute the following command:
java -Xmx10g -cp *baseDir*/lib/sqlite-jdbc-3.23.1.jar;uva-diffusion-model-1.0.jar uva.nssac.model.DiffusionModeller *baseDir*/input-data/svr/svr-input.properties
On Linux/Unix operating system execute the following command:
java -Xmx10g -cp *baseDir*/lib/sqlite-jdbc-3.23.1.jar:uva-diffusion-model-1.0.jar uva.nssac.model.DiffusionModeller *baseDir*/input-data/svr/svr-input.properties

4 Input Data

4.0.1 *Input property file description*

Description of useful parameters in the input property file.
region : Name of the region (e.g. svr)
baseDir: Project checkout folder path. This acts as the base directory.
modelVersion : Model number
inputCSV : Name of the input csv file with household records.
hidMapperFile : Mapping file of synthetic household identifier to another series. It is needed by our models but not relevant to the user.
mile1File : 1-mile network file location
mile3File : 3-mile network file location
mile4File : 4-mile network file location
networkDBPath : Households network graphs database path
modelPropertyPath : Property file path for the model under consideration.
outFile : Filename at the end of each iteration
aggregatedOutFile : Output of all the iteration is aggregated in this file.
replicates : Number of times the simulation should run with same parameter setting
ticks : Timesteps in the simulation (e.g. 10).

incentive : Do you want to apply incentive (true/false)
seedingStrategy : random (one seeding strategy implemented)
seedingTicks : At which time points in the simulation should free solar be allocated (e.g. 3 or 2,5,8)
totalBudget : Total number of households in the region that will receive free solar panels.

4.0.2 *Household Records*

This is the csv file containing household records with all the required parameters for the model except the household neighborhood features.

4.0.3 *Household Neighborhood Data*

3 synthetic household networks are generated for 1-mile, 3-mile, and 4-mile neighbors. Each of these networks is stored in an adjacency list.

4.1 Output Data

1. CSV Files are generated per replicate :: ReplicateNo, Tick, CumulativeAdopters

2. Single file are generated at the end of the simulation to aggregate the output of each replicate :: Tick, Average,%Adopters, LowerBound, UpperBound, StdDev, Min, Max, [Adoptions per replicate,,,,]
   where,
   Tick = Timestep number in the simulation.
   Average = No. of adopters averaged over all the simulation replicate runs.
   %Adopters = Percentage of adopters in the region.
   LowerBound = Calculated as (stdDev-Average)
   UpperBound = Calculated as (stdDev+Average)
   StdDev = Standard deviation across number of replicates
   Min = Minimum number of adopters across the replicates
   Max = Maximum number of adopters across the replicates

   All other column header values that are numbered are the replicate run numbers and their respective adoptions per tick.

The location of all these files is present in the input files.