ROSE (Rough Outlier Set Extraction) v3.0

ROSE is a machine learning algorithm that implements outlier detection of an unlabeled spatiotemporal dataset using a rough set approach. It also provides a representative subset of the original data, describing the same structure, with which it is possible to detect the same outliers, named kernel set.

The paper that reports all the details and should be cited when the code is used is

Albanese A, Sankar K P, Petrosino A., IEEE Transactions on Knowledge and Data Engineering, Vol. 26, no. 1, pp. 194-207, 2014, DOI: 10.1109/TKDE.2012.234.

ROSE is written in Java and can be downloaded here . It is platform independent and bundled as jar package. Java Runtime Environment (JRE) is needed to run this software.

ROSE package content:

- ROSE.jar (java executable file)
- README.md
- LICENSE
- sample_dataset.txt
- Original paper 2014_Albanese_Sankar_Petrosino.pdf
- Supplementary material folder ttk2014010194

ROSE can be executed from a Unix-like shell, a Terminal for Mac-OS or Windows CMD as follows:

```
java -jar ROSE.jar <path input dataset> <number of outliers> <number of neighbors> <number of elements once> <alpha>
```

This is an example using the bundled sample dataset (assuming you are in the ROSE root folder):

```
java -jar ROSE.jar sample_dataset.txt 16 16 18 1
```

Input parameters

- path input dataset - Input dataset filename (complete absolute path). This has to be a text file where each row contains numerical values, ranging in [0, 1] (normalized). Each numeric value must be space separated (two or three space characters are allowed). These values are the coordinates of the instances to be processed.

  Each row specifies:

- required configuration: *(x,y,t)* triplets where *(x,y)* is a spatial coordinate and *(t)* is a timestamp.

  Example:

```
0.84506904 0.10248589 0.15307587
      0.84543708 0.10339779 0.15308264
      0.84524589 0.10418321 0.15308941
      0.84540431 0.10549813 0.15309618
      0.84588570 0.10667086 0.15310294
```

- optional configuration: 13 values are allowed at most, (x,y,t) + (f1,...,f10) where the first 3 values are inherited from required configuration and *(f1,...,f10)* is a list of other features.

  Example:

```
0.84506904 0.10248589 0.15307587 0.84588570 0.10667086 0.15310294
      0.84543708 0.10339779 0.15308264 0.84524589 0.10418321 0.15308941
      0.84540431 0.10549813 0.15309618 0.84588570 0.10667086 0.15310294
```

- standard configuration: not all 13 values need to be specified and just some features are specified (the sample dataset only contains *(x,y,t)*).

- number of outliers - the number of outliers to be extracted from the data set.
- number of neighbors - the number of neighbors to be considered.
- number of elements once - it is the number of items that are extracted from the input data set at each iteration (and then compared with the rest of the file). This must be greater than the number of outliers. number of elements once > number of outliers
- alpha - is a multiplier value of the linear combination of the weights, ranging between *[0,1]* (for spatial weight) and
1 - alpha

(for temporal weight).

Special cases: alpha = 1 only spatial components are considered. alpha = 0 only temporal component is considered.

Output data

Each execution of the ROSE software will produce an output folder named as the dataset input file name. Each execution on the same input dataset will overwrite the existing output folder.

- TopOutlier-*numoutlier*-*alpha*-*weight*-*milliseconds*.txt - (*milliseconds* value in the filename is used to maintain processing history, TopOutlier-
inputfilename
-
numoutliers
-
alpha
-
weight
.txt is the most recent iteration). These files contain TopOutliers search results at each iteration. The last two files contain the Lower and Upper approximation of the outlier set respectively.
- WeightedData-*inputfilename*-*numoutliers*-*weight*.txt - contains the starting points set with weights for each point.
- SolvingSet-*milliseconds*.txt - contains the kernel set (*milliseconds* value in the filename is used to maintain processing history, SolvingSet.txt is the most recent iteration).
- NegativeRegion.txt - contains discarded instances.