NESTOREv1.0 functions

In the following, we show in detail the general workflow of the single modules in terms of functions and sub-functions.

Cluster identification module

Run_clus_identification.m is the main script of the module, which identifies clusters from the seismicity of a specific area using a window-based method. The user can make several choices, such as space-time window equations for aftershocks and foreshocks, minimum magnitude for mainshock, and input and output file names. The main cluster identification functions are:

1. **input_clus_id.m**: It takes input parameters from input file "fileinput_clus_identification.txt" as the name of catalog, the minimum value for cluster mainshock magnitude Mm, the magnitude-dependent relationships for spatial and temporal selection of aftershocks/mainshocks of cluster.
2. **clus_ident.m** it provides a list of mainshocks, aftershocks, foreshocks and a set of additional information, such as the temporal distance between the mainshock and the strongest aftershock, the number of aftershocks, the magnitude of completeness M_c and its uncertainty for each cluster. Several subfunctions of clus_ident.m were originally part of the Zmap software (Wiemer 2001) and, in some cases, have been updated and improved for the NESTOREv1.0 implementation (further details are available in the function comments or in the readme file)

Training Module

Run_training.m is the main script of the module. It reads the output file provided by the cluster identification module and extracts features from the clusters, obtaining appropriate thresholds and statistical information for them. The user can make several choices, such as the region and the time periods in which the training should be performed, the maximum depth, the completeness magnitude, the time intervals T_i in which the analysis should be done, and the input and output file names.

The training procedure is a two-step process. In the first stage, the features and their performance are calculated for each period T_i. In the second phase, only the values and thresholds corresponding to the "good interval" are taken into account. To clearly distinguish the first and second parts, the words "nostop" and "stop" have been included in the function names, respectively. The main training functions are:

1. **input_train.m**: It takes input parameters from input file "fileinput_training.txt" as the name of clusters catalog, the name to be used for the output files, the name of file containing the coordinates of the studied area, the maximum depth for the mainshock, the starting and ending time of the analysis, the completeness magnitude, and the vector of analysis periods T_i.
2. **feacalc_nostop.m:** evaluates features for each time period T_i.

3. **check_T_clus.m:** disables the processing for time periods T_i in which the number of clusters is smaller than 10 or when the number of type A clusters is smaller than 2.
4. **perf_loo_nostop:** evaluates the performance of the features for all the time periods T_i.
5. **showperf_th05_nostop.m:** reads the performance files (Accuracy, Precision, Recall, Informedness), shows their trend and finds the "good interval".
6. **rebuild_feafile_stop.m:** rewrites the features only for the periods belonging to good interval and inherits the last value of the interval for the longer time intervals.
7. **perf_loo_stop.m:** evaluates the performances of the new features file.
8. **showperf_stop.m:** reads the performance files of the previous step and plots the trend of performance parameters.
9. **ROC_diag.m:** plots ROC diagrams for each period T_i.
10. **threshfinder.m:** calculates the threshold for features in each period T_i belonging to the good interval.
11. **calcprobA_th05.m:** estimates the probability to be a type A cluster under and over the feature thresholds, basing on the number of A and B clusters in the corresponding period T_i.

Testing module

Run_testing.m is the main script of this module; it uses the thresholds and statistical information provided by the training module to classify the clusters of the test set; by comparing the true class of the clusters with the result of the classification, it provides statistical information on the quality of the classification.

In order to fully exploit the information obtained from the training part, the region corresponding to the clusters of the test set should be the same as the training region. The testing procedure only uses features and threshold values referring to reliable intervals (good interval) found by the training module. Therefore, the feature names of this module only contain the word "stop".

The main subprograms of the testing module are:

1. **input_test.m**: it takes the input parameters from the input file “fileinput_testing.txt” as the name of clusters catalog, the files’ name related to training output, the name to be used for the output files, the name of file containing the coordinates of the studied area, the maximum depth for the mainshock, the starting and ending time of the analysis, the completeness magnitude, and the vector of analysis periods.
2. **feacalc_stop.m:** evaluates the features for each time period T_i using the good interval found in the training module.

3. **class_stop.m:** using the output files from the training module, obtains the classification for each feature and merges it by eq. (3) to obtain the final NESTOREv1.0 classification. Writes the results to the classification files.
4. **class_perf.m**: evaluates the performance of the features using the classification files.
5. **class_bayes_fig.m**: plots the probability of being a type A cluster for each cluster in the test set and for increasing values of T_i.
6. **check_T_clus.m:** disables the processing for time periods T_i in which the number of clusters is smaller than 9 or when the number of type A clusters is smaller than 2.
7. **ROC_class_diag.m:** illustrates the performance using the ROC diagram and the Precision-Recall diagram.

Near-real-time classification module

Run_nrt_class.m is the main script of this module; starting from the input catalog containing the ongoing cluster, identifies its earthquakes, extracts the corresponding features and, by using the thresholds and statistical information calculated by the training module for the same area, estimates the probability during time that the cluster is of class A.

Most of the subfunctions of the near-real-time classification module are shared with the cluster identification and testing modules. The main subfunctions are:

1. **input_nrt.m**: it takes input parameters from input file “fileinput_nrt_class.txt” as the name of catalog for the occurring sequence, the minimum magnitude for cluster mainshock, the magnitude-dependent relationships for the spatial and temporal

selection of aftershocks/mainshocks of cluster, the files' name related to the training output, the name to be used for the output files, the name of the file containing the coordinates of the studied area, the maximum depth for the mainshock, the starting and ending time of the analysis, the completeness magnitude, the vector of analysis periods, Mm and the last analysis period the user wants to consider.

2. **clus_ident.m**: as in the Cluster Identification Module.
3. **nrt_maps.m:** plots the distribution of the magnitude as function of time and a map of the aftershocks' epicentres centered on the mainshock one for the ongoing cluster.
4. **feacalc_stop.m:** as in Testing Module.
5. **nrt_class_stop.m:** from the output files of the training module obtains the classification for each feature and, by using a Bayesian approach, merges it by eq. (3) to obtain the near-real-time classification for the cluster. It produces a figure for each analysis period showing the probability to be a type A cluster estimated for each feature. In addition, it shows the probability to be a type A as a function of T_i up to the last period selected by the user and, according to its estimation on the all T_i, provides the ATLS classification for the ongoing cluster.
6. **result_msg.m:** shows the overall result of classification and the time and space windows of the forecasting.