JPCE

A Java implementation of Projective Clustering Ensembles

April 15, 2011

Welcome. The Projective Clustering Ensembles project is based on the work:

> F. Gullo, C. Domeniconi, A. Tagarelli (2011) *Advancing Data Clustering via Projective Clustering Ensembles*. In Proc. ACM International Conference on Management of Data (SIGMOD'11), Athens, Greece, June 12-16, 2011.

JPCE is a Java software package that implements PCE. This package is free for research, academic and non-profit making purposes only. If you use this piece of software for your work and got something published please include the above citation. The software may not be sold or redistributed without prior approval. One may make copies of the software for their use provided that the copies, are not sold or distributed, are used under the same terms and conditions. As unestablished research software, this code is provided on an "as is" basis without warranty of any kind, either expressed or implied. The downloading, or executing any part of this software constitutes an implicit agreement to these terms. These terms and conditions are subject to change at any time without prior notice.
The JPCE package can be downloaded from http://people.dimes.unical.it/andreatagarelli/software/jpce.

SYSTEM REQUIREMENTS

JPCE is written in Java (**Java(TM) Platform, Standard Edition, Runtime Environment, Version 6**) and has been tested under Windows XP and Mac OS X.

PROJECT STRUCTURE

JPCE is provided as a .zip archive. Once decompressed, the JPCE project folder contains 5 files:

- **JPCE_README.pdf**
- **JPCE.jar**
- **start.sh**
- **start.bat**
- **parameters.properties**

and three subfolders:

- **raw_datasets**
- **ensembles**
- **results**

Only the **ensembles** subfolder is currently empty. However, it can be filled with the ensembles used in the paper by downloading the **SIGMOD11_RWE_Ensembles*.rar** archives from http://people.dimes.unical.it/andreatagarelli/software/jpce/jpce-ensemble-datasets/ and decompressing such archives directly into the subfolder. See the description of the **ensembles** subfolder provided next for further details.

A description of each of the files and subfolders is reported next.

Files

* File **JPCE.pdf** --- this file.

* File **JPCE.jar** --- this is the Java archive containing all Java classes of the JPCE project.

* Files **start.sh** and **start.bat** --- these are the usual files to run JPCE on Unix and Windows platforms, respectively. Either allows the setting of the path to find the local JRE installation (the JRE directory in case of **start.bat**), and the max/min memory heap size before invoking **JPCE.jar**.

* File **parameters.properties** --- this file lists the parameters needed by JPCE; it may contain up to 20 <parameter-name> = <parameter-value> pairs (look at the **parameters.properties** file currently stored into the project folder for an example). The parameters are logically grouped into two categories, each one corresponding to a different task that JPCE may perform.

- *Ensemble generation* task, which performs the generation of the ensembles according to the methodology described in Section 4.1.1 of the paper:
    - `do_ensemble_generation` (valid values: Boolean; optional; default value: false) --- enables/disables the generation of the ensembles.

    - `datasets_in_ensemble_generation` (valid values: strings; required only if `do_ensemble_generation =` true) --- the name(s) of the input data for the ensemble generation task; if more values are specified, these are listed separated by commas. The data names must be consistent with prefixes of the names of the data files stored in the **raw_datasets** folder. Example: if `datasets_in_ensemble_generation =` dataset1,dataset2, the **raw_datasets** folder must contain the files *dataset1.data* and *dataset2.data.*
    - `number_of_ensembles_in_ensemble_generation` (valid values: integers greater than 0; optional; default value: 20) --- the number of ensembles to be generated for each dataset specified in the `datasets_in_ensemble_generation` parameter. A value equal to N for this parameter will lead to the generation of the <dataset-name>_ENSEMBLE_X.data files in the **ensembles** folder, for each X = 1,...,N, for each <dataset-name> specified in the `datasets_in_ensemble_generation` parameter.
    - `ensemble_size_in_ensemble_generation` (valid values: integers greater than 1; optional; default value: 200) --- the number of clustering solutions in each ensemble to be generated.
- *PCE* task, which runs and evaluates the various PCE methods:
    - `do_PCE` (valid values: Boolean; optional; default value: true) --- enables/disables the execution of the PCE task (as subsequent to the ensemble generation task); if true, it is assumed the presence of the ensemble data files in the folder **ensembles**
    - `datasets_in_PCE` (valid values: strings; required) --- the name(s) of the input data for the PCE task; if more values are specified, these are listed separated by commas. The data names must be consistent with prefixes of the names of the data files stored in the **raw_datasets** folder (see `datasets_in_ensemble_generation` parameter).
    - `number_of_ensembles_in_PCE` (valid values: integers greater than 0; optional; default value: 1) --- the number of ensembles used in the PCE task for each dataset specified in the `datasets_in_PCE` parameter. A value equal to N for this parameter assumes the presence of the <dataset-name>_ENSEMBLE_X.data files in the **ensembles** folder, for each X = 1,...,N, for each <dataset-name> specified in the `datasets_in_PCE` parameter.
    - `PCE_algorithms` (available values: MOEA-PCE,EM-PCE,CB-PCE,FCB-PCE; required only if `do_PCE =` true) --- specifies the PCE algorithm(s) to apply; if more values are specified, these are listed separated by commas.
    - `number_of_runs` (valid values: integers greater than 0; optional; default value: 50) --- the number of runs to be performed, for each dataset, ensemble, and PCE algorithm.
    - `number_of_clusters_in_consensus_clusterings` (valid values: integers greater than 1; optional; default value: the number of classes of the reference classification (if it exists) for each dataset) --- the number(s) of clusters in the consensus clustering(s) outputted by each PCE algorithm, for each dataset. Exactly one value for

each dataset specified in the `datasets_in_PCE` parameter is required. Multiple values are listed separated by commas and follow the same ordering of the datasets in the `datasets_in_PCE` parameter.

- `evaluation_wrt_reference_classification` (valid values: Boolean; optional; default value: true) --- enables/disables the evaluation of the clustering results wrt the available reference classification(s) of the dataset(s) (see Sections 4.1.3 and 4.2.1 of the paper). If true, the **Evaluation w.r.t. reference classification*.csv** files (described below) are generated into the **results** folder. This parameter is discarded (i.e., it is implicitly assumed to be false) if no reference classification is available with the input dataset(s) on which the PCE task is performed.
- `evaluation_wrt_ensemble_solutions` (valid values: Boolean; optional; default value: true) --- enables/disables the evaluation of the clustering results wrt the ensemble solutions (see Sections 4.1.3 and 4.2.1 of the paper). If true, the **Evaluation w.r.t. ensemble solutions*.csv** files (described below) are generated into the **results** folder.
- `save_consensus_clusterings` (valid values: Boolean; optional; default value: false) --- enables/disables the saving of the consensus clusterings outputted by each PCE algorithm, for each dataset, ensemble, and run. If true, the consensus clusterings are stored into the **ConsensusClustering*.data** file, whose format is described later in this document.
- `MOEA-PCE_population_size` (valid values: integers greater than 0; optional; default value: 30) --- the population size parameter (*t*) of the MOEA-PCE algorithm (see Section 4.1.2 of the paper). This parameter is ignored if MOEA-PCE has not been specified into the `PCE_algorithms` parameter.
- `MOEA-PCE_max_iterations` (valid values: integers greater than 0; optional; default value: 200) --- the number of maximum iterations (*I*) of the MOEA-PCE algorithm (see Section 4.1.2 of the paper). This parameter is ignored if MOEA-PCE has not been specified into the `PCE_algorithms` parameter.
- `EM-PCE_alpha_parameter` (valid values: integers greater than 1; optional; default value: 2) --- the value of the \alpha parameter in the EM-PCE algorithm (see Section 4.1.2 of the paper). This parameter is ignored if EM-PCE has not been specified into the `PCE_algorithms` parameter.
- `CB-PCE_alpha_parameter` (valid values: integers greater than 1; optional; default value: 2) --- the value of the \alpha parameter in the CB-PCE algorithm (see Sections 3.2.1 and 4.1.2 of the paper). This parameter is ignored if CB-PCE has not been specified into the `PCE_algorithms` parameter.
- `CB-PCE_beta_parameter` (valid values: integers greater than 1; otpional; default value: 2) --- the value of the \beta parameter in the CB-PCE algorithm (see Sections 3.2.1 and 4.1.2 of the paper). This parameter is ignored if CB-PCE has not been specified into the `PCE_algorithms` parameter.

- `FCB-PCE_alpha_parameter` (valid values: integers greater than 1; optional; default value: 2) --- the value of the \alpha parameter in the FCB-PCE algorithm (see Sections 3.2.1 and 4.1.2 of the paper). This parameter is ignored if FCB-PCE has not been specified into the `PCE_algorithms` parameter.
- `FCB-PCE_beta_parameter` (valid values: integers greater than 1; optional; default value: 2) --- the value of the \beta parameter in the FCB-PCE algorithm (see Sections 3.2.1 and 4.1.2 of the paper). This parameter is ignored if FCB-PCE has not been specified into the `PCE_algorithms` parameter.

Subfolders

* Subfolder **raw_datasets** --- contains the input dataset files (.data). Currently, it contains the 10 datasets used for the experimental evaluation presented in the paper (i.e., iris, wine, glass, ecoli, yeast, segmentation, abalone, letter, tracedata, controlchart, see Section 4.1) which have the following format: each line corresponds to an object and contains numerical values separated by a semicolon. The first value in the line denotes the ID of a class (in the reference classification), and the subsequent values denote the object's attribute (feature) values. Class IDs are integer progressive values starting from 0; if no reference classification is available, all lines begin with the same class ID (e.g., 0).
To carry out experiments on different dataset(s), it is sufficient to put the corresponding .data file(s) conforming to the above format into the **raw_datasets** folder.

* Subfolder **ensembles** --- contains the files storing the ensembles for each input dataset (files named as <dataset-name>_ENSEMBLE_<#Ensemble>.data, where <#Ensemble> denotes the ensemble number represented as a progressive integer value starting from 1). This is the output folder of the *ensemble generation* task of JPCE. Each file in this folder stores one ensemble for a particular dataset and conforms the following format. It begins with the string "*# Clustering:*" followed by the number of clustering solution in the ensemble in the next line, and is organized in as many blocks as the number of clustering solutions contained in the ensemble (i.e., the ensemble size). Each of these blocks has the following format:

- the block begins with three values (one per line): the number of clusters, the Boolean value stating if the object-to-cluster assignment is hard (false) or soft (true), and the Boolean value stating if the feature-to-object assignment is equally-weighted (false) or unequally-weighted (true) (see Section 2.2 in the paper).

- for each cluster, two lines are reported: the first corresponds to the object-to-cluster assignments (referred to as \Gamma values in the paper, see Section 2.2), whereas the second corresponds to the feature-to-cluster assignments (referred to as \Delta values in the paper, see Section 2.2).

This folder is currently empty. To perform experiments, one has three choices:

1. download the **SIGMOD11_RWE_Ensembles*.rar** archives containing all the ensembles used for the experimental evaluation presented in the paper (20 ensembles for each dataset within the **raw_datasets** folder) from http://people.dimes.unical.it/andreatagarelli/software/jpce/jpce-ensemble-datasets/; currently, two archives are available:

SIGMOD11_RWE_Ensembles_SmallDatasets.rar, which contains the ensembles for iris, wine, glass, ecoli, yeast, segmentation, tracedata, and controlchart datasets (this archive is provided as divided into two volumes, each one of about 300 MB), and **SIGMOD11_RWE_Ensembles_LargeDatasets.rar**, which contains the ensembles for abalone, and letter-recognition datasets (this archive is provided as divided into four volumes, each one of about 300 MB);

2. generate the ensembles according to the methodology presented in the paper (see Section 4.1.1); this can be carried out following the description of the *ensemble generation* task provided above;
3. perform experiments on its own generated ensemble(s); in this case, it is sufficient to put the corresponding .data file(s) conforming to the above rules into the **ensembles** folder.

* Subfolder **results** --- contains the clustering result and run log files outputted by each execution of JPCE. These files are organized in four types (look at the files currently contained into the **results** folder for an example of each possible output file) :

- **Evaluation*.csv** --- reports the accuracy clustering results in CSV format. This can be directly converted in a tabular format to look like Tables 3-4 in the paper. Note that each execution of JPCE will produce up to 6 **Evaluation*.csv** files, i.e., 3 concerning the evaluation w.r.t. the reference classification (**Evaluation w.r.t. reference classification*.csv** files) and 3 concerning the evaluation w.r.t. the ensemble solutions (**Evaluation w.r.t. ensemble solutions*.csv** files), depending on the Boolean values of the `evaluation_wrt_reference_classification` and `evaluation_wrt_ensemble_solutions` parameters. The details of the assessment criteria used for the evaluations are reported into Section 4.1.3 of the paper.
- **ConsensusClustering*.data** --- reports the consensus clustering solutions outputted by each PCE algorithm specified into the `PCE_algorithms` parameter. For each dataset, ensemble, PCE algorithm, and run (in this order), the corresponding consensus clustering is represented by two lines for each of its clusters, where the first line corresponds to the object-to-cluster assignments (referred to as \Gamma values in the paper, see Section 2.2) and the second corresponds to the feature-to-cluster assignments (referred to as \Delta values in the paper, see Section 2.2). Note that, (only) in case of MOEA-PCE, more clusterings are reported for each run as such an algorithm outputs multiple consensus clusterings; in particular, the number of output consensus clusterings by MOEA-PCE for each run is exactly equal to the population size *t* specified in the `MOEA-PCE_population_size` parameter. Note that the **ConsensusClustering*.data** file is created depending on the Boolean value of the `save_consensus_clusterings` parameter.

  This file can be exploited, for instance, for evaluating the consensus clusterings outputted by the various PCE algorithms according to measures different to those used in the paper.
- **Execution times*.csv** --- reports the time performance (in milliseconds) in CSV format. It tells about the times of each PCE algorithm on each dataset, averaged over the ensembles and the runs. This file can be directly converted in a tabular format to look like Table 5 in the paper.

- **SIGMOD11*.log** --- reports the complete log of the executions of JPCE on all datasets, and for all ensembles, algorithms and runs. Looking at the log file is the only way for checking the progress in test execution, as no other feedback is provided to the user. This log is structured as follows:
    a. a block reporting the summary of all parameters in the **parameters.properties** file;
    b. a block for each dataset reporting a notification of the ensembles generated (this block is written only if the `do_ensemble_generation` parameter is true);
    c. a block for each dataset reporting the count and name of the dataset, along with the number of objects, attributes, and classes (this block is written only if the `do_PCE` parameter is true);
    d. within each block of dataset, a block for each ensemble reporting the count and size of the ensemble;
    e. within each block of ensemble, a block for each algorithm reporting the parameter values used with the algorithm and the count of runs completed.

INSTRUCTIONS

1. Decompress the JPCE.zip archive.

2. Locate into the **raw_datasets** and **ensembles** subfolders the .data files you want to use for the experiments according to the instructions provided above (optional).

3. Open the **start.sh** (**start.bat**) file in the main folder with a text editor, set therein the path to find the local JRE installation, and (optionally) set the max/min memory heap size. The path of the local JRE installation (including the \bin subfolder) must be written right after the “JRE_HOME=” string; the minimum memory heap size (in MB) to be used must be written right after the “XMS_DEFAULT=” string; the maximum memory heap size (in MB) to be used must be written right after the “XMX_DEFAULT=” string.
   Example (windows): if your local JRE installation folder is “C:\Programs\jre” and you want to set 1024 MB and 4096 MB for minimum and maximum memory heap size, respectively, the first three lines in the **run.bat** file must be as follows:

   ```
   set JRE_HOME="C:\Programs\jre\bin"
   set XMS_DEFAULT=1024
   set XMX_DEFAULT=4096
   ```

4. Run **start.sh** (**start.bat**)... enjoy!