S. Iovleff
P. Bathia
C. Biernacki

# I Coding conventions

For the coding we use the java coding convention. All data members are defined with an underscore at the end of their name, like `data_`.

Data members referring to pointer start with `p_`

All alternatives should be enclosed in `enum`.

The notations proposed in this document can be modified if needed or enclosed in namespace in order to avoid name collision.

# II Components of the library

## 1 Algorithms

The algorithms will be :

```
enum Algo
{
  em_,
  cem_,
  sem_,
  mcem_,
  exact_
}
```

The priority is to implement the `*em` versions, the `exact_` is just there in case of, but should not be implemented. Should we add stochastic minimization algorithms ?

## 2 Stopping criteria

The stopping criteria will be :

```
enum StopCriteria
{
  deltaLnLikelihood_,
  deltaPostProbabilities_,
  deltaParameters_,
  nbIterMax_
}
```

It should be possible to mix different criteria, for example `deltaLnLikelihood_|nbIterMax_` mean that we want to stop the iteration when one of the criteria is true.

## 3 Data initialization

The initialization of the algorithm will be :

```
enum Initialization
{
  randomPartition_,
  randomParameters_,
  givenPartition_,
  givenPosteriorProbabiblity_,
  givenParameters_
}
```

Did i forget a method for initialization? The random cases are the priority. In case of heterogeneous distributions it will be difficult to find a way for initialization with given parameters.

## 4 Strategies

A strategy is composed of three parts :

1. multiple initializations,
2. for each initialization a short run,
3. A long run.

### a short runs

A short run will be an arbitrary number of sequence `(Algo_, StopCriteria)`. For example :

```
{(sem_, 1000), (cem_, 0.01|1000)}, {(em, 0.01)}
```

mean that in a short run, there is 1000 iterations of the SEM algorithm, and at most 1000 iterations of CEM that will be stopped if some other criterion have a delta less than 0.01 and finally iterations of the EM until the delta of some criteria is less than 0.01.

### b long run

A long run is initialized with the better of the short run and a StopCriteria. If there is no short run, with an initialization.

## 5 Model Selection Criteria

The model criteria will be :

```
enum Criteria
{
  aic_,
  bic_,
  icl_,
  cv_,
  penExtern_
}
```

It should be possible to let an user to define its own penalization criteria (`penExtern_` option). Should we add cross-validation ?

# III Plugin specifications (Parmeet)