FisPro 3.5

fispro@supagro.inra.fr

January 2013

`http://www.inra.fr/mia/M/fispro/`

This version has been tested with Windows 7 64 bits, Windows XP 32 bits, Linux 32 and 64 bits. In case of trouble during install or at runtime, do not hesitate to report them to us. This is important to improve FisPro quality and to help other users.

1 New in FisPro3.5

Two main new functionalities in version 3.5 :

1.1 Optimization module

The optimization module provides a cross validation procedure. Pairs of learning/test samples are generated. For each pair, a FIS is designed from the learning sample and its accuracy is calculated on the corresponding test sample.

All of the optimized FIS are then aggregated into a final unique FIS. Each paramer is the median of the corresponding parameters in the optimized FIS. Generally the final FIS average accuracy on test samples is better than the average of optimized FIS accuracies. A detailed presentation can be found in

Serge Guillaume and Brigitte Charnomordic. Parameter optimization of a fuzzy inference system using the fispro open source software . In IEEE Catalog Number: CFP12FUZ-USB, editor, IEEE International Conference on Fuzzy Sys tems, pages 402-409, Brisbane, Australia, June 2012. IEEE.

1.2 Generation of a distance matrix

Fuzzy partitions can be used to generate a matrix of distances between individuals of a data set. It allows to introduce expert knowledge in distance computation.

Such a distance is called FP-based distance. The program generates a (n x n) matrix of the distances between the (n) rows of the current data file. In the multivariate case, it is possible to combine Euclidean distances and FP-based distances in the different input dimensions. The aggregation is done with a Minkowski-like combination. The distance matrix can be exported to be used in clustering algorithms by other programs, such as R.

A detailed presentation can be found in:

Serge Guillaume, Brigitte Charnomordic, and Patrice Loisel. Fuzzy partitions: a way to integrate expert knowledge into distance calculations. International Journal of Information Sciences, page In Press, 2012.

The functionalities are available in the Distance option of the Data menu.

1.3 Other changes

- Documentation :

  The guide *Quickstart with FisPro* is now available in Spanish.

- Interface : on Windows, automatic association with FisPro of the files with the fis extension (the association is done in the registry during installation)

- bug correction:

  on Windows: each new window now opens on the foreground.

- Partition quality indices:

  Two indices are associated to each fuzzy partition, the partition coefficient, PC, to be maximized, and the partition entropy, PE, to be minimzed:

  In the following formulae, c is the number of MFs, n the number of rows in the data set and u_{ik} the membership degree of the kth example in the ith MF.

$$PC = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^2}{n}$$

$$PE = -\frac{1}{n}\left\{\sum_{k=1}^{n}\sum_{i=1}^{c}\left[u_{ik}\log_a(u_{ik})\right]\right\}.$$

- Distance within the fuzzy partition,:

  FISIN::Distance, can now be normalized (default value), or not.

- HFP Module:
  - An implementation error of the ’symbnum’ option has been fixed, which makes it competitive with the ’numerical’ option.
  - The partition initialization is now done with triangular edges instead of the semi-trapezoidal ones, which gave too much weight to extreme values.
  - In the HFP configuration file, a few heading parameters have been removed (given here with their default value) :

    ```
      Simplified distance=0
      Distance MF Nb=-1
      Variance MF Nb=999
      MergingCriterion=’distance’
      HeteroPenalty=0.000000
      HeteroProportion=0.500000
    ```

    The corresponding functions have also been removed :

    ```
    hfpfis.h
      static const char *MergingDistance() { return "distance"; }
      static const char *MergingVariance() { return "variance"; }
      char *GetMerging() { return strMergingCrit; }
      void SetMerging(const char *merging);
      int GetDistanceMFNb() { return NmfDist; }
      void SetDistanceMFNb( int distance ) { NmfDist = distance;
      int GetVarianceMFNb() { return NmfVar; }
      void SetVarianceMFNb( int variance ) { NmfVar = variance; }
      int GetSimplifiedDistance() { return SimpDist; }
      void SetSimplifiedDistance( int simplified ) { SimpDist = s
      double GetHeteroPenalty() { return HetPenal; }
      void SetHeteroPenalty( double penalty ) { HetPenal = penalt
      double GetHeteroProportion() { return HetProp; }
      void SetHeteroProportion( double proportion ) { HetProp = p

    hfpin.h
      void Init4MF(int s, int before, int \& mfmin, int \& mfmax)
      void Init3MF(int s, int before, int \& mfmin, int \& mfmax)
      void Init2MF(int s, int before, int \& mfmin, int \& mfmax)
      void SimpliDist(int smin, int smax);
      void Density(double *Dens);
      double Variance(double *Y, double *VarSef);
    ```

```
        void VarianceNP(double *Y, int e, double \&v, double \&p);
```

Many thanks to FisPro users for their help: bug detection and correction, compilation on various platforms or with various compilers, translation.

2 Known problems

2.1 Installing from a deb package

The following procedure is required to access the download repository.

- **Graphical mode, with the Synaptic package manager**

  Configuration -> repository :

  + Other software -> Add:

  deb http://download.opensuse.orgrepositorieshome:misteaxUbuntu_10.04 .

  + Authentification -> Import the key

  Fetch the file Release.key

  Reload repositories

  Install the FisPro package

- **Command line**

  sudo echo "deb http://download.opensuse.org/repositories/home:/mistea/xUbuntu_10.04 ./" | tee -a /etc/apt/sources.list

  sudo apt-get update

  sudo apt-get install fispro

  eventually add the key (available on FisPro web site)

  sudo apt-key add Release.key

2.2 Installing from a rpm package - CentOs 32 bits

The following error message "cannot restore segment prot after reloc" appears on some CentOs distributions when running FisPro installed from the rpm binary package. The reason is that the fispro rpm provides a shared object library. A workaround is to disable SElinux. For permanent solution, login as root, edit the /etc/selinux/config to set SELINUX=disabled on the file. Do a reboot after editing the file.

2.3 Java 64 bits and win 32 binary

The following error message ‘Can’t load IA 32-bit .dll on a AMD 64-bit’ appears when running a 32 bit version of FisPro, if java 64 bit is installed on a Windows 7 64 bit system. A solution is to uninstall java 64 bit, and install a java machine from www.java.com, or else to install a 64 bit version of FisPro.

3 Note for programmers and command line mode

- **All learning programs, when called from command line, have a -wl option, that allows a silent (wordless) mode. Only error messages will appear on the output. This option overrides the -a option.**
- Working directory

  The C++ binaries create files in the directory they are run from.
- C++ binaries command line options are not yet harmonized. The *loopoptim* and *loopoptimsample* programs require a space between the option name and its argument, for instance:

  loopoptim rice.fis rice optimfis 10 0.005 1000 1000 0.000001 -in ’1 4 3’ -r -m 0.10 -c 0.10

  Please refer to the *Learning* documentation for examples.