Akademie věd České republiky
Ústav teorie informace a automatizace, v.v.i.

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Kamil Dedecius

Implementation of partial forgetting in Mixtools

No. 2284 October 26, 2010

ÚTIA AV ČR, P.O.Box 18, 182 08 Prague, Czech Republic
Tel: +420 286892337, Fax: +420 266052068, Url: http://www.utia.cas.cz,
E-mail: dedecius@utia.cas.cz

1 Introduction

This report describes the implementation of the partial forgetting in the Mixtools software package. The related theory of the partial forgetting, can be found in [1].

2 Usage

The partial forgetting for projection-Bayes estimation of ARX mixture is called using the `mixestimpf` function, based on the original `mixestim` function. Its definition is as follows:

```
function mixestimpf (in  Mix,
                     in  weights,
                     in  ndat,
                     in  MixAlt,
                     in  method,
                     in  weightsForg,
                     in  selftune,
                     in  compForg,
                     in  threshold
)
```

There are several input parameters:

```
% Full version with threshold
Mix = mixestimpf(Mix, weights, ndat, MixAlt, method, weightsForg, \
                 selftune, compForg, threshold)

% forgetting factor for components' weights (default 1.0)
Mix = mixestimpf(Mix0, weights, ndat, MixAlt, method, weightsForg, \
                 selftune, compForg)

% selftuning of weights (default 1)
Mix = mixestimpf(Mix0, weights, ndat, MixAlt, method, weightsForg, \
                 selftune)

% forgetting of weights of partial forgetting
Mix = mixestimpf(Mix0, weights, ndat, MixAlt, method, weightsForg)

% method (default projection Bayes)
Mix = mixestimpf(Mix0, weights, ndat, MixAlt, method)

% partial forgetting with defined alternative mixture
Mix = mixestimpf(Mix0, weights, ndat, MixAlt)

% MixAlt = Mix0 flattened by factor 0.99
Mix = mixestimpf(Mix0, weights, ndat)
```

```
% recursive estimation
Mix = mixestimpf(Mix0, weights)

% estimation without forgetting
Mix = mixestimpf(Mix0)
```

Parameters:

`Mix0`	initial mixture, any type
`weights`	partial forgetting weights
`ndat`	scalar containing number of data samples to be processed or a pair containing data filename and rowcount of data stored in the file
`MixAlt`	alternative mixture for partial forgetting
`method`	estimation method (cf. mixestim)
`weightsForg`	forg. factor for weights of partial forgetting
`selftune`	selftuning of forgetting weights
`compForg`	forgetting of components
`threshold`	threshold for estimation

Return values:

`Mix`	estimated mixture

See also:

`mixestim`

USE:

Say that we have a mixture with 2 factors, each having n=2 parameters. The total number of hypotheses of partial forgetting is $2^n = 4$ hypotheses. Then we define the weights of these hypotheses as:

```
weightsFactor1 = [0.1, 0.2, 0.3, .4]
weightsFactor2 = [0.1, 0.2, 0.0, .7]
weightsForg = {weightsForg1, weightsForg2}
```

The structure of `weightsFactor(1)` follows from this simple rule: Let 0 = constant parameter, 1 = varying parameters. Let's have a matrix with alternating 2^i-tuples in i-th column; $i = 1, \ldots, n$. I.e.:

```
[1, 1]
[0, 1]
[1, 0]
[0, 0]
```

Hence `weightsFactor1` means: Both parameters vary (`[1,1]`) with probability 0.1 Second parameter varies (`[0,1]`) with probability 0.2 First parameter varies (`[1,0]`) with probability 0.3 Both parameters are constant (`[0,0]`) with probability 0.6. Try `permutationmatrix(2)` in Matlab.

- If `MixAlt` in not defined, then posterior flattened with factor 0.99 is used;
- if `weightsForg` is not defined, the default value for `compForg` is used;
- if `selftune` is 1, then weights are tuned according to likelihoods, otherwise weights are constant.

3 Additional information

The `mixestimpf` function calls these two self-standing auxiliary functions:

- `permutationmatrix` – its purpose is explained above. It generates a matrix with 0's and 1's with all existing permutations.
- `genmixpf` – creates a pseudomixture for approximation.

In addition, the following functions are defined in `mixfrgpf.m`:

- `reweight` – data and time update of hypotheses' weights.
- `approx` – approximation of the mixtures by a single pdf.

Acknowledgement

This work was supported by grant GAČR 102/08/0567 Fully probabilistic design of dynamic decision strategies and by the Research center DAR, project of MŠMT 1M0572.

References

[1] Dedecius, K. et al. (2010). *Parameter Tracking with Partial Forgetting Method.* Submitted to International Journal of Adaptive Control and Signal Processing.