

ReportMCMC クラスの使い方

2008 年 11 月 22 日

1. 準備.

- ファイル ReportMCMC.h, ReportMCMC.ox をプログラムファイルと同じフォルダか, Ox の include フォルダにいれ, ReportMCMC.ox を読み込むためにプログラムファイルに #include <ReportMCMC.ox> という文を追加する.

2. 使い方. オブジェクトの引数として MCMC の標本系列を用いて ReportMCMC(標本系列) とする. 例えば MCMC により 1,000 個の標本が, (β_1, β_2) という 2 つのパラメータについて得られており, それぞれ vbeta1_s, vbeta2_s という 2 つの 1000×1 ベクトルに保存されているとする. オブジェクトの名前を out とすると,

```
decl out = new ReportMCMC(vbeta1_s~vbeta2_s);
out.Report();
delete out;
```

とすれば, (β_1, β_2) に関する結果 (事後分布の要約統計量, 事後密度関数・標本自己相関係数・標本経路の図など) が出力される.

3. 使うことができるメンバ関数とその出力内容.

以下では特にオプションを用いない限り, デフォルトでは変数の名前は Param1, Param2, ... とし, ファイル名の先頭ラベルは out, 図の種類を表す拡張子は eps となっている.

- CalculateStatistics(). 事後分布の平均, 標準偏差, 95%信用区間, 中央値, Geweke の収束判定の p 値 (帰無仮説:収束), 非効率性因子 (inefficiency factor). 変数間の分散共分散と相関係数の行列 (下三角: 共分散, 対角要素: 分散, 上三角: 相関係数).
- DrawDensityPlot(). 事後確率密度関数の図を出力.
出力ファイルの名前は「“先頭ラベル”_Density_it_“MCMCの反復回数”.“図の種類”」.
- DrawACFPlot(). 標本の自己相関関数の図を出力.
出力ファイルの名前は「“先頭ラベル”_ACF_it_“MCMCの反復回数”.“図の種類”」.
- DrawSamplePath(). 標本経路の図を出力.
出力ファイルの名前は「“先頭ラベル”_SamplePath_it_“MCMCの反復回数”.“図の種類”」.
- OutputFile(). 結果のファイルへの出力.
出力ファイルの名前は「“先頭ラベル”_Samples_it_“MCMCの反復回数”.dat」.
- Report(). 上のすべてを行う.

4. オプション

- `SetBandwidth(const dBm)`. バンド幅の設定. 標本時系列について平均の標準誤差を求めるときなどに使うバンド幅を `dBm` に変更できる. (デフォルトでは MCMC 反復回数の 10% に設定).
- `SetFormat(const sformat)`. `format` コマンドと同じ. 出力の列が長いときなどに使う.
- `SetFigureType(const sfigtype)`. 図の種類を指定. デフォルトは `eps`.
- `SetOutfileName(const soutfilename)`. 出力ファイル名の先頭ラベルを指定. デフォルトは `out`.
- `SetVarNames(const asname)`. 出力ファイルの中の変数ラベルを指定. デフォルトは `Param1, Param2, ...`.
- `SetCorrOutput(const fcorr)`. パラメータの標本相関係数・共分散行列の出力を指定. 出力するときは `fcorr=1`, 出力しないときには `fcorr=0`. デフォルトは出力.
- `SetMaxNumFig(const cmaxnumfig)`. 複数の図を出力する際に 1 ページあたりに同時出力する最大個数. 最大個数を 4 としたいときには出力するときは `cmaxnumfig=4`. デフォルトは `cmaxnumfig=16`.

使用例.

```
decl out = new ReportMCMC(vbeta1_s~vbeta2_s);
out.SetBandwidth(100);           // バンド幅を 100 に変更
out.SetFormat(300);             // 出力の列数を 300 に変更
out.SetFigureType("gwg");       // 図の形式を gwg に変更
out.SetOutfileName("myout");    // ファイル名の先頭ラベルを myout に変更
out.SetVarNames({"beta1","beta2"}); // 変数名を beta1, beta2 に変更
out.Report();
delete out;
```

5. その他のメンバ関数.

以下では `mX` は $n \times K$ の行列で, n は MCMC の反復回数, K は変数の個数, 各列は各変数の標本であるとする. また `dBm` はバンド幅とする.

- `fTsvaer(const mX, const dBm)`. 標本平均の分散の推定値を計算. スペクトル密度関数を用いる.
- `fTsvaer_Batch(const mX, const dBm)`. 標本平均の分散の推定値を計算. バッチ平均を用いて計算する.

- `fGeweke(const mX, const dBm)`. Geweke の収束判定を行い, p 値を計算する. 帰無仮説は収束.
- `fCI(const mX)`. 95%信用区間が 0 を含むときは 0, 含まないときは 1, さらに 99%信用区間が 0 を含まないときは 2 を返す.

6. 改訂.

- Geweke (1992) による収束検定統計量のラベルを CD に変更. また, 標本平均の分散の推定値を計算する際のバンド幅を $2\sqrt{n_i}$ ($i = 1, 2$) (n_1, n_2 はそれぞれ標本数) に変更. 2008 年 11 月 22 日.
- Inefficiency のラベルを IF に更新. 2008 年 11 月 22 日.
- ヘッダファイルのタイプを訂正. 2008 年 11 月 22 日.