Users' Guide for HRM

Jie Chen & Xiaoming Huo

September 18, 2008

1 Main Matlab M file regarding HRM

The following is a list of main functions in the HRM simulations:

- `mainfunc_converge.m`: simulations regarding the convergence theorem.
- `mainfunc_GCV.m`: simulations regarding generalized cross validation (GCV) principle and MSE.
- `mainfunc_pred1.m`: one-step and iterative two-step predictions with HRM for synthetic examples.
- `mainfunc_pred2.m`: direct two-step prediction with HRM for synthetic examples.
- `ex_sunspot.m`: prediction with HRM for the sunspot example.
- `ex_blowfly.m`: prediction with HRM for the blowfly example.
- `sunspot.mat` and `blowfly.dat` provide the data for sunspot example and blowfly example.

See Appendix A for other supporting functions and their calling protocols. In Matlab, you can always use "help FUNCTION-NAME" to view the header of our files, which explain their functionality.

2 R Codes for Other Methods

The software for FAR was downloaded from

> http://orfe.princeton.edu/∼jqfan/fan/nls.html.

Implementation of TAR and AAR is based on an online software package that is downloadable at

> http://cran.r-project.org/src/contrib/Descriptions/tsDyn.html. (Maintainer: Antonio, Fabio Di Narzo.)

Implementation of Locpoly is based on an online software package that is downloadable at

> http://cran.r-project.org/src/contrib/Descriptions/JLLprod.html. (Maintainer: David Tomás, Jacho-Chávez.)

Implementation of AR can be found in the Matlab system identification toolbox , and Loess is implemented based on the function "loess" in standard R package "stats".

2.1 Simulations regarding prediction with synthetic examples

`datagenerate300iter.m` generates a data file for 300 iterations. `AAR_R.R` and `TAR_R.R` illustrate the implementation of AAR and TAR. `FAR_1R.R` is the code for the one-step and iterative two-step prediction using FAR, and `FAR_2R.R` is the code for the direct two-step prediction using FAR. `Locpoly_R.R` and `Loess_R.R` illustrate the implementation of Locpoly and Loess.

2.2 Blowfly example about prediction

`blowfly_FAR.R` and `blowfly_TAR.R` are the codes of FAR and TAR for blowfly example.

3 Conclusion

Please refer to the web site www.isye.gatech.edu/∼xiaoming/software for related and up-to-date information.

A Matlab Functions regarding HRM

These functions are used to implement HRM.

A.1 datageneration()

Description

Generate time series examples according to option

Usage

```
[Xt,Xttrue,error]=datageneration(NTS0, sigma, option)
```

Input

`NTS0`	length of intended time series
`sigma`	standard deviation of the noise
`option`	method of data generation; option =1,2,3,...

Output

`Xt`	time series with length=NTS0, i.e., Xt in the model $X_t = f(Z_t) + \varepsilon$
`Xtrue`	expected value of the time series, i.e. $f(Z_t)$ in the model $X_t = f(Z_t) + \varepsilon$
`error`	noise, i.e., ε in the model $X_t = f(Z_t) + \varepsilon$

A.2 ZYgeneration()

Description

Generate response Y and exploratory variable Z_t from the time series X_t with lag p for one-step prediction and iterative two-step prediction

Usage

```
[Zt,Y]=ZYgeneration(Xt, p)
```

Input

`Xt`	time series
`p`	time lag of the time series X_t

Output

`Zt`	exploratory variable
`Y`	response variable

A.3 ZYgeneration_direct()

Description

Generate response Y and exploratory variable Z_t from the time series X_t with lag p for direct two-step prediction

Usage

```
[Zt,Y]=ZYgeneration(Xt, p)
```

Input

`Xt`	time series
`p`	time lag of the time series X_t

Output

`Zt`	exploratory variable
`Y`	response variable

A.4 GenerateMmatrix()

Description

Generate the M matrix that is described in the HRM paper

Usage

```
[M]=GenerateMmatrix(Zt, K, p)
```

Input

`Zt`	the exploratory variable that is generated from the timesseries X_t
`K`	the number of nearest neighbors used to calulate M
`p`	time lag of the time series

Output

`M`	M matrix that is used in the HRM model, where n-t-s stands for nonlinear time series

A.5 GCV4()

Description

Calculate the generalized cross validation (GCV) function

Usage

```
[GCV]=GCV3(lambda,Y,M)
```

Input

`lambda`	tuning parameter in the HRM model
`Y`	response variable generated from the time series X_t
`M`	M matrix that is used in the HRM model

Output

`GCV`	GCV function

A.6 computefhat()

Description

Calculate the denoised Y with formula $(I + \lambda M)^{-1}$

Usage

```
fhat=computefhat(M,Y,lambda)
```

Input

`M`	M matrix that is used in the HRM model
`Y`	response variable generated from the time series X_t
`lambda`	tuning parameter in the HRM model

Output

`fhat`	the denoised Y with formula $(I + \lambda M)^{-1}$

A.7 predictstep()

Description

one-step prediction

Usage

```
[fpred]=predictstep(fhat, Zt, Z, K, p, option)
```

Input

`fhat`	the denoised Y with formula $(I + \lambda M)^{-1}$
`Zt`	the exploratory variable that is generated from the timesseries X_t
`Z`	the exploratory variable for prediction
`K`	the number of nearest neighbors for prediction
`p`	time lag of the time series
`option`	choose the prediction method. There are two choices: 1. option=1: first order prediction 2. option=2: second order prediction

Output

`fpred`	one-step prediction

A.8 vectorize1

Description

Suppose size$(A) = [n, n]$, then we calulate the vector $B = (A_{11}, A_{22}, \cdots, A_{nn}, \sqrt{2}A_{12}, \cdots, \sqrt{2}A_{1n}, \sqrt{2}A_{23}, ..., \sqrt{2}A_{2n}, \cdots, \sqrt{2}A_{n-1,n})$

Usage

```
B=vectorize1(A)
```

Input

`A`	square matrix

Output

`B`	vector which is equal to $(A_{11}, A_{22}, \cdots, A_{nn}, \sqrt{2}A_{12}, \cdots, \sqrt{2}A_{1n}, \sqrt{2}A_{23}, ..., \sqrt{2}A_{2n}, \cdots, \sqrt{2}A_{n-1,n})$

A.9 checkConvergence()

Description

check the convergence of the theoretical bound

Usage

```
[C,F,an,bn,gamma,lambda]=checkConvergence(M, f, error,sigma)
```

Input

`M`	M matrix that is used in the HRM model
`f`	vector of the true values of function $f(\cdot)$.
`error`	noise, i.e., ε in the model $X_t = f(Z_t) + \varepsilon$
`sigma`	standard deviation of the noise

Output

`an and bn`	quantities a_n and b_n that are defined in the manuscript
`gamma and lambda`	quantities γ_n and λ_n that are defined in the manuscript
`C`	$C_n\sigma^2$
`F`	$(Uf)^2$

References

[1] J. Chen and X. Huo. A hessian regularized nonlinear time series model. Technical report, Georgia Tech., December 2007. downloadable at www2.isye.gatech.edu/statistics/papers/08-01.pdf.