

ECONO – biblioteca de functii pentru analiza seriilor de timp

Ec. Adrian HOSPODAR

Articolul prezinta biblioteca ECONO, destinata studierii seriilor de timp unidimensionale. În introducerea se prezinta câteva din caracteristicile bibliotecii ECONO. Clasa de baza a bibliotecii este prezentata pe larg în următoarele secțiuni ale articolului. Modalitatea de utilizare a funcțiilor oferite de către această bibliotecă este demonstrată cu ajutorul unui exemplu simplu.

Cuvinte cheie: biblioteca, funcție, colecție, metodă.

1. Introducere

Biblioteca ECONO oferă accesul la o colecție de proceduri și algoritmi utilizați în analiza econometrică a seriilor de timp. Principala sa arie de aplicare este, până acum, analiza seriilor de timp unidimensionale. Funcțiile oferite de bibliotecă sunt:

a) *Funcții ce implementează tehnici de transformare inițială a datelor.* În utilizarea acestor funcții există posibilitatea de a acorda controlul asupra alegerii parametrilor, ordinii de aplicare a mai multor tehnici diferite etc., fie utilizatorului fie programului. Cazul în care controlul aparține utilizatorului va fi numit *funcționare asistată*, celalalt caz *funcționare neasistată*.

b) *Funcții de calcul a unor proprietăți statistice ale seriei de timp* (autocorelație, autocorelație parțială etc.).

c) *Funcții ce specifică un model teoretic de aproximare a dinamicii seriei de timp.* În acest moment, modelele teoretice utilizate sunt modelele din clasa modelelor liniare, cunoscute sub denumirea generală ARMA, și cele din clasa modelelor neliniare în medie. Datorită concepției întregii biblioteci, prezentată în următoarea secțiune, extinderea gamei de modele este o sarcină facilă. Referitor la utilizarea funcțiilor de specificare a modelelor teoretice, trebuie spus că ele pot fi folosite atât asistat, cât și neasistat. Funcționarea neasistată se bazează pe existența regulilor de decizie pe baza cărora se selectează un anumit tip de model. Controlul procesului

de specificare este trecut de la utilizator către program. Această modalitate se înscrie în tendințele prezente observate pe piața pachetelor software dedicate analizelor econometrice.

d) *Funcții ce selectează un tip de model teoretic.* Precedenta categorie de funcții ofereau drept principală funcționalitate specificarea ordinului (ordinelor) și/sau parametrilor unui model teoretic, tipul modelului teoretic fiind deja stabilit. Funcțiile din această categorie au drept scop tocmai alegerea tipului de model teoretic. Pentru aceasta, două abordări diferite sunt folosite. Prima se bazează pe utilizarea testelor statistice pentru selecția unui anumit tip de model teoretic. Cea de a doua se bazează pe evaluarea rezultatelor obținute în urma aplicării unui sau mai multor modele teoretice. Pe baza acestor evaluări se alege tipul de model ce aproximează cel mai bine tipul de dinamică prezentat de o anumită serie de timp.

e) *Funcții ce evaluează rezultatele procesului de modelare.* Evaluarea se face analizând proprietățile statistice ale valorilor reziduale obținute din estimarea unui tip de model teoretic.

f) *Funcții ce construiesc portofolii de modele diferite.* Abordarea teoretică utilizată pentru îndeplinirea acestui obiectiv este analiza bayesiană.

Organizarea bibliotecii ECONO este determinată de modalitatea de proiectare utilizată. Aceasta este cea orientată obiect. ECONO este compusă din mai multe clase.

Relatiile de mostenire existente între aceste clase conduc la formarea unei ierarhii a claselor. Aceasta ierarhie este discutata în sectiunea 3.

ECONO se prezinta sub forma unui fisier de tip *dynamic link library*. Utilizarea functionalitatilor bibliotecii într-un program propriu se poate face în doua moduri. Prima modalitate este de a folosi clasele componente ale bibliotecii, care sunt exportate. Avantajele acestei modalitati este accesul la structuri de date, algoritmi si proceduri deja implementate. Dezavantajul major pare a fi scaderea flexibilitatii programului ce beneficiaza de functiile bibliotecii.

Cea de a doua modalitate este de a folosi functiile interne ale claselor componente ale bibliotecii (acestea sunt si ele exportate, deci pot fi accesate). Fiecare metoda a unei clase componente nu face altceva decat sa apeleze o functie interna. Functia efectueaza prelucrarile iar rezultatul returnat de catre functie este returnat si de catre metoda. Diferenta dintre metoda si functie este ca metoda se aplica doar asupra unei instante a clasei, iar rezultatele sunt memorate de regula în membri ai clasei iar functia nu tine cont decât de tipul de data al argumentului/argumentelor si al rezultatului. Acesta face ca functia sa poata fi utilizata în conditii mult mai generale decât metoda.

Functionalitatile bibliotecii se reduc, în acest moment, la efectuari de calcule. Ea nu ofera functii proprii de realizare a unor grafice. Totusi, rezultatele prelucrarilor efectuate pot fi vizualizate grafic prin utilizarea unor alte programe utilitare sau biblioteci. De aceea, oferirea de functii grafice proprii nu reprezinta în acest moment o necesitate.

Biblioteca ECONO beneficiaza la rândul ei de serviciile unei alte biblioteci, denumita NewMat. Aceasta efectueaza toate calculele cu matrici. Versiunea utilizata a acestei biblioteci este 09. Biblioteca este oferita în regim freeware. Sursele si docu-

mentatia acestei biblioteci pot fi gasite la adresa <http://webnz.com/robert/>.

2. Evolutia bibliotecii ECONO

Proiectul de dezvoltare al bibliotecii ECONO a început în noiembrie 1998, ca parte a contractului de cercetare *Metode de analiza si predictie a comportamentului întreprinderilor mici si mijlocii*. În prima faza, ea a fost conceputa ca un pachet software de sine statator. Dupa realizarea unui prim prototip, s-a considerat ca efortul de întreținere si dezvoltare a unui întreg pachet software este prea ridicat.

Începând cu ianuarie 1999, prototipul initial a fost modificat pentru a fi adus în forma unei biblioteci cu legaturi dinamice. Prima versiune a acestei biblioteci a fost realizata în martie 1999. Modificarile survenite de atunci s-au regasit într-o a doua varianta, definitivata în iunie 1999.

Începând cu cea de a doua varianta a bibliotecii ECONO a început si scrierea documentatiei pentru functiile oferite de catre biblioteca. Versiunea 2.04 este prima versiune complet documentata. Documentatia are forma unui fisier *rich text format*. Un fisier help cu același continut este în curs de definitivare.

Biblioteca ECONO a fost dezvoltata la început utilizând un compilator Gnu sub sistemul de operare Linux. Prototipul programului si primele versiuni ale bibliotecii au fost realizate utilizând acest mediu de dezvoltare. Versiunea 2.00 a bibliotecii ECONO a fost compilata si sub sistemul de operare Windows. Compilatorul folosit a fost Visual C++ 4.00. Începând cu aceasta versiune s-a renuntat la realizarea de versiuni pentru Linux. Toate versiunile ulterioare au fost realizate utilizând Visual C++ 4.00 pentru Windows. Respectarea standardelor ANSI asigura însa ca realizarea de versiuni pentru Linux (sau alte sisteme de operare) se poate face cu efort minim pentru orice versiune a bibliotecii.

Biblioteca pentru manipularea matricilor NewMat a fost utilizata ca biblioteca statica pentru versiunile bibliotecii ECONO mai mici decât 2.02. Începând cu aceasta versiune, ea a fost compilata sub forma unei biblioteci cu legaturi dinamice. Acest lucru a permis reducerea dimensiunii produsului finit.

Începând cu versiunea 2.03 s-a realizat si un kit de instalare pentru biblioteca ECONO. Acest kit de instalare usureaza instalarea pe alte calculatoare.

3. Ierarhia claselor

Biblioteca ECONO în versiunea 2.04 contine urmatoarele clase:

1. CTmeS. Aceasta este clasa de baza a bibliotecii. Ea ofera functii pentru calculul unor proprietati statistice ale seriei de timp si pentru transformarea initiala a datelor. Transformarea initiala a datelor se poate face asistat sau neasistat.
2. CArma. Contine functii de transformare preliminara a datelor pentru mode-

larea prin modele ARMA, functii de specificare si estimare a modelelor si functii de estimare a calitatii procesului de modelare.

3. CRezid. Contine functii ce calculeaza proprietati statistice specifice valorilor reziduale ce rezulta din estimarea unui model teoretic.

4. CTar. Contine functii de specificare si estimare a modelelor din clasa TAR si functii de estimare a calitatii procesului de modelare.

5. CBayes. Contine functii de calcul a verosimilitatilor marginale si de construire a portofoliilor de modele.

4. Clasa CTmeS

Pentru clasa CTmeS se vor prezenta membrii si metodele componente. Deoarece semnificatia membrilor clasei este evidenta se va face doar un scurt comentariu pentru fiecare. Metodele care prezinta un interes particular vor fi prezentate pe larg.

Tabel 1. Membrii clasei CTmeS

Membri publici	Tip data	Comentarii
Abatere_moment	Double	Retine valoarea abaterii momentului centrat de ordin 3
Date observate	Pointer la double	Retine datele componente ale seriei de timp
Diferenta	Integer	Numarul de diferentieri necesare pentru stationarizarea seriei
Dispersia	Double	Retine valoarea dispersiei seriei de timp
Fi	Double	Retine valoarea parametrului fi cu care se apeleaza metodele ExponentDeCorelatie si IntegralDeCorelatie
Integral	Double	Retine valoarea integralei seriei de timp
IntegralAbsoluta	Double	Retine valoarea integralei absolute a seriei de timp
IntegralDeCorelatie	Double	Retine valoarea integralei de corelatie a seriei de timp
Media	Double	Retine valoarea mediei seriei de timp
Nr_elemente	Integer	Retine numarul de termeni ai seriei de timp
Nume_fisier	Pointer la char	Retine numele fisierului ce contine valorile initiale ale seriei de timp observate
Membri privati		
StructMemorie	Structura	Memoreaza adresa la care se memoreaza membrul DateObservate.

Membrii clasei CTmeS au fost selectati astfel încât sa conduca la eficientizarea calculelor ulterioare. Eficientizarea s-a facut prin prisma a doua criterii: timpul de calcul si spatiul de memorie necesar pentru serializarea obiectelor clasei. Pe aceste considerente s-au inclus printre membrii

clasei marimi ca integrala de corelatie, a carei calculare este o sarcina ce necesita mult timp de calcul. Marimi ca autocorelatia sau autocorelatia partiala, a caror calculare este suficient de rapida dar pot ocupa un spatiu considerabil de memorie, nu se regasesc însă printre membrii clasei.

Tabel 2 Metodele clasei CtmeS

Metode	Tip data argumente	Tip data rezultat	Comentarii
AbatereMoment()	void	Double	Calculeaza momentul centrat de ordin 3 pentru DateObserve
AlegeCox_Box()	void	Double	Alege parametrul λ pentru transformarea Cox-Box corespunzatoare DateObserve(vezi mai jos)
AlegeDif()	void	Int	Alege numarul de diferentieri necesar pentru stationarizarea DateObserve
AutoCorelatie()	Pointer la double	Int	Calculeaza primele k valori ale functiei de autocorelatie pentru DateObserve
AutoCP()	Int	Pointer la double	Calculeaza primele k valori ale functiei de autocorelatie partiala pentru DateObserve
Cox_Box()	Pointer la double	Pointer la double	Aplica transformarea Cox-Box asupra DateObserve
Diferentiere()	Pointer la double	Pointer la double	Diferentiaza de un numar finit de ori seria de timp primita ca argument al metodei.
Dipsersie()	void	Double	Calculeaza dispersia pentru DateObserve
Exponent de corelatie	(double, int, int)	double	Calculeaza exponentul de corelatie pentru DateObserve
IncarcaDate()	Pointer la double	Void	
Integrala()	void	Double	Calculeaza integrala pentru DateObserve
IntegralAbsoluta()	void	Double	Calculeaza integrala absoluta pentru DateObserve
IntegralDeCorelatie()	double	Double	Calculeaza integrala de corelatie pentru DateObserve
Medie()	void	Double	Calculeaza media pentru DateObserve

Metoda IncarcaDate()

Deoarece clasa este exportata folosind macroul *DllExport* constructorul si destructorul clasei nu sunt declarati explicit, ci generati de catre compilator. Aceasta creste portabilitatea bibliotecii în diverse medii de programare. Metoda *IncarcaDate()*, desi nu este constructor al clasei, ofera aceeasi functionalitate. Constructorul generat de catre compilator nu face altceva decât sa rezerve spatiu pentru membrii clasei. Metoda *IncarcaDate()* copiaza valorile seriei de timp în membrul *Date* Observate si calculeaza (sau initializeaza) urmatorii membri ai clasei: *Nr_elemente*,

$$x_t = a_{k1} * x_{t-1} + a_{k2} * x_{t-2} + \dots + a_{kk} * x_{t-k} + u_t$$

Pentru un proces autoregresiv de ordin k coeficientul a_{kk} se calculeaza astfel:

$$a_{kk} = \frac{\begin{vmatrix} 1 & r_1 & \dots & r_{k-2} & r_1 \\ r_1 & 1 & & r_{k-3} & r_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{k-1} & r_{k-2} & \dots & r_1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & r_1 & \dots & r_{k-2} & r_{k-1} \\ r_1 & 1 & & r_{k-3} & r_{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{k-1} & r_{k-2} & \dots & r_1 & 1 \end{vmatrix}}$$

unde prin r_k s-a notat valoarea autocorelatiei de ordin k .

Pentru o serie de timp, autocorelatia partiala a seriei se calculeaza folosind ecuatiile recursive:

$$a_{kk} = \frac{r_k - \sum_{j=1}^{k-1} a_{k-1,j} * r_{k-j}}{1 - \sum_{j=1}^{k-1} a_{k-1,j} * r_j}; \quad a_{kj} = a_{k-1,j} - a_{kk} * a_{k-1,k-j} \quad j=1,2,\dots,k-1$$

unde prin r_k s-a notat autocorelatia de ordin k .

Metoda *AutoCP()* foloseste aceste ecuatii recursive pentru calculul autocorelatiilor partiale. Deoarece metoda foloseste valorile autocorelatiei este recomandat ca ea sa fie apelata dupa apelarea metodei *AutoCorelatie()*. De obicei, aceasta ordine de apelare corespunde si secventei ce trebuie urmată pentru modelarea seriei de timp.

Metoda AlegeDif()

Metoda returneaza numarul de ori de care este necesara aplicarea operatorului de

Abatere_Moment, Dispersia, Media, Integral, IntegralAbsoluta. Acestia sunt utilizati frecvent de catre celelalte metode ale clasei.

Metoda AutoCP()

Metoda este apelata cu parametrul l , de tip integer. Acesta specifica întârzierea maxima pentru care se calculeaza autocorelatia partiala. Metoda returneaza un vector de dimensiuni $(l \times 1)$ ce contine valorile autocorelatiei partiale. Autocorelatia partiala se defineste teoretic ca fiind coeficientul a_{kk} din autoregresia de ordin k :

diferentiere pentru stationarizarea seriei. Acest parametru este ales astfel încât sa minimizeze dispersia seriei de timp rezultata prin diferentiere.

Metoda AlegeCox_Box()

Metoda determina valoarea parametrului λ

din transformarea Cox-Box $g(x_t) = \frac{x_t^l - 1}{l}$,

unde prin x_t s-au notat termenii seriei initiale.

Parametrul λ este ales astfel încât să minimizeze abaterea distribuției seriei transformate față de distribuția normală.

Utilizarea celor două metode pentru stationarizarea unei serii de timp este prezentată în Listingul 1.

```
CTmeS SeriaMea; //declara variabila SeriaMea ca obiect CtmeS
SeriaMea.Nume_fisier="d:\date\test.dat"; //Datele initiale sunt in fisierul 'test.dat'
SeriaMea.IncarcaDate(); //citeste datele din fisier si calculeaza primii indicatori statistici;
SeriaMea.Diferentiere(AlegeDif()); //diferentiaza seria pentru a o stationariza in medie.
//Numarul de ori de care se diferentiaza seria este dat de valoarea returnata de AlegeDif()
SeriaMea.Cox_Box(AlegeCox_Box()); //aplica transformarea Cox-Box cu parametrul  $\lambda$  //dat
de metoda AlegeCox_Box()
```

Listing 1 Stationarizarea unei serii de timp folosind funcțiile clasei CtmeS

Rezultatul dat de secvența de cod de mai sus este o serie stationarizată în medie și dispersie. Aceasta poate fi modelată în continuare cu un model ARMA, folosind metodele clasei CARMA. Pentru exemplificarea secvenței de cod prezentată în Listingul 1 se va folosi o serie de timp de dimensiuni reduse. Aceasta reprezintă totalul creditelor acordate în sistemul bancar românesc în intervalul martie 1993 – iunie 1996. Datele analizate au fost exprimate în preturile corespunzătoare lunii martie 1993. Ele sunt reprezentate

grafic în figura 1, împreună cu cele exprimate în preturi nominale. Stationarizarea datelor s-a făcut prin utilizarea funcțiilor bibliotecii ECONO. Pentru aceasta, datele au fost diferențiate o singură dată pentru a fi stationare în medie. Stationarizarea în dispersie s-a făcut prin utilizarea transformării Cox-Box, cu parametrul λ egal cu 0.28. Datele astfel transformate sunt reprezentate grafic în figura 2.

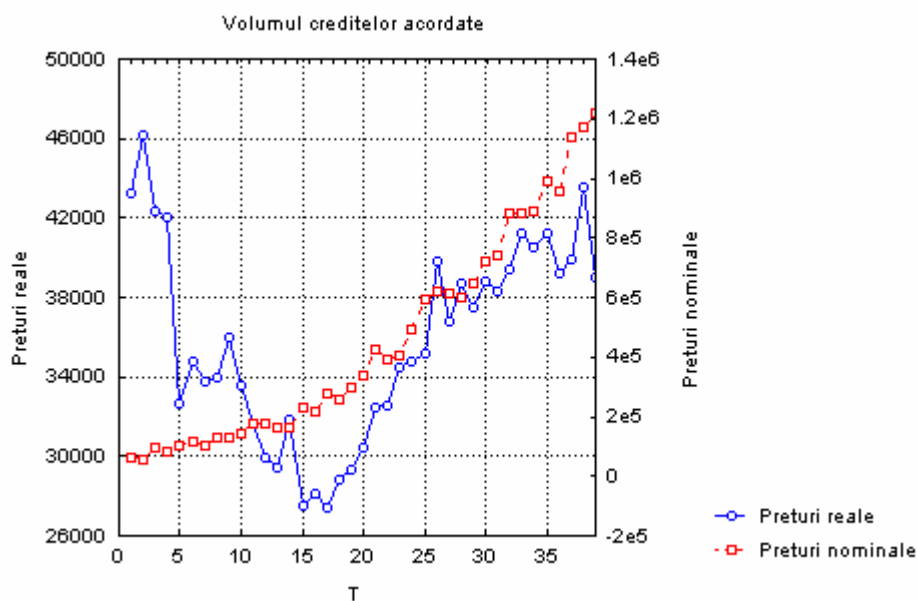


Figura 1 Reprezentarea grafică a creditelor acordate în sistemul bancar românesc (exprimate în preturi reale și nominale)

