

ECONOMETRIE NON LINEAIRE

Présentation du cours

M2 Economie - Université Paris-Dauphine

- Marie Bessec (marie.bessec@dauphine.psl.eu)
- Département d'économie (LEDa) à Dauphine

Présentation générale du cours

- Cours d'économétrie des séries temporelles où l'on présentera deux grandes classes de modèles à changement de régimes : les modèles à seuil et les modèles à changement de régime markovien et leur application à des séries macroéconomiques et financières.
- On présentera également certains outils utilisés en *machine learning* (les arbres de régression et les forêts aléatoires) qui ont une certaine parenté avec les modèles à seuil.

Présentation générale du cours (suite)

Les outils présentés sont utiles pour répondre aux questions que se posent les économistes, comme par exemple :

- La dynamique des prix des actifs est-elle similaire en cas de forte/faible volatilité des marchés financiers ?
- Les marchés financiers réagissent-ils identiquement à des hausses ou à des baisses du prix du pétrole ?
- Y a-t-il une bulle spéculative dans les prix de l'immobilier parisien ?
- Les marchés financiers sont-ils haussiers ou baissiers ?
- L'économie française est-elle dans un état conjoncturel favorable ou défavorable ?
- Quel sera le taux de croissance du PIB au trimestre prochain ? Quelle sera la consommation d'électricité entre 8h et 9h demain ?

Présentation générale (suite)

- Organisation des séances : 2 séances 1/2 théoriques, 4 séances 1/2 avec applications sur Matlab.
- Installation de Matlab sur vos ordinateurs
<https://my.dauphine.fr/page/services-et-salles/logiciels>
- Matériel de cours : polycopié avec transparents de cours et sujet des TP
- Evaluation : examen en salle informatique

Sessions théoriques

- 1 Introduction sur les modèles à changement de régime
- 2 Rappel sur le modèle AR linéaire
- 3 Les modèles AR à seuil
- 4 Les modèles AR à changement de régime markovien
- 5 Machine learning : arbre de régression et forêt aléatoire

Sessions informatiques

- 1 TP1- Simulation d'un processus AR, estimation et test
- 2 TP2- Simulation d'un processus SETAR, estimation et test
- 3 TP3- Simulation d'un processus MS-AR, estimation et test
- 4 TP4- Arbre de régression et forêt aléatoire

- Franses P.H., van Dijk D. (2000), Non-linear time series models in empirical finance, Cambridge University Press.
- Hamilton, J. D., Time Series Analysis, Princeton : Princeton University Press, 1994.
- Hastie, T., Tibshirani, R., & Friedman, J. 2009. In : The Elements of Statistical Learning : Data Mining, Inference, and Prediction. Springer-Verlag.

ECONOMETRIE NON LINEAIRE

Chapitre 1. Introduction sur les modèles à changements de régime

M2 Economie - Université Paris-Dauphine

Modèle linéaire versus MCR

Soit Y_t et X_t deux variables observées sur $t = 1, \dots, T$.

- **Modèle linéaire** : la relation entre Y_t et X_t est la même à tous les instants du temps.

$$Y_t = c + \beta X_t + \sigma \varepsilon_t \quad t = 1, \dots, T \quad (1)$$

avec ε_t iid $(0, 1)$

ou de façon équivalente :

$$Y_t = c + \beta X_t + u_t \quad t = 1, \dots, T \quad (2)$$

avec u_t iid $(0, \sigma^2)$

Modèle linéaire versus MCR

- Supposons maintenant qu'il existe plusieurs **états** ou **régimes**
- Exemples
 - ▷ expansion versus récession
 - ▷ saison chaude ou froide
 - ▷ marchés financiers haussiers/baissiers, volatils ou non
 - ▷ régimes de politique monétaire (accomodante/restrictive)
 - ▷ état sanitaire de la population
- La relation entre Y_t et X_t peut différer suivant l'état dans lequel se trouve le système considéré

Modèle linéaire versus MCR

On définit S_t une variable d'état égale à 1 si on est dans le régime 1 en t et 2 si on est dans le régime 2 en t .

- **Modèle à changement de régime** : La relation entre Y_t et X_t diffère suivant le régime en t

$$Y_t = \begin{cases} c_1 + \beta_1 X_t + \sigma_1 \varepsilon_t & \text{si } S_t = 1 \\ c_2 + \beta_2 X_t + \sigma_2 \varepsilon_t & \text{si } S_t = 2 \end{cases} \quad t = 1, \dots, T \quad (3)$$

avec ε_t iid (0, 1). On écrit de façon plus synthétique :

$$Y_t = c_{S_t} + \beta_{S_t} X_t + \sigma_{S_t} \varepsilon_t \quad t = 1, \dots, T \quad (4)$$

avec ε_t iid (0, 1).

Modèle linéaire versus MCR

- **Remarque** : le modèle linéaire est un cas particulier du modèle à changement de régime. Si $c_1 = c_2$, $\beta_1 = \beta_2$ et $\sigma_1 = \sigma_2$, on retrouve le modèle linéaire. Pour tester l'existence de changements de régime, il conviendra donc de tester la validité de ces contraintes.

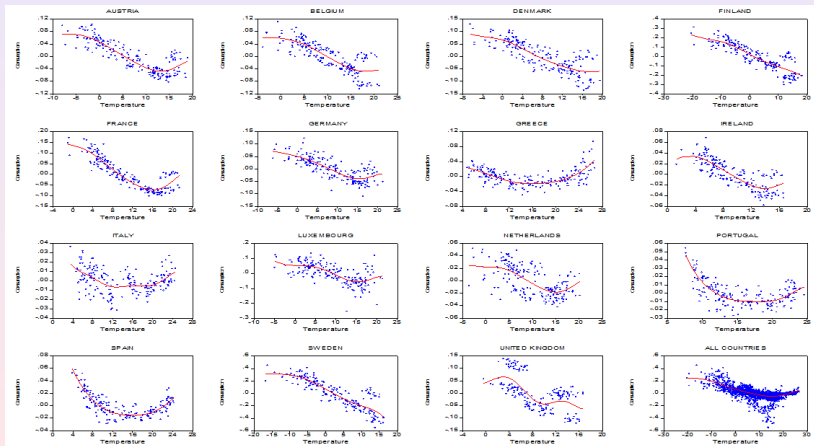
Modèle à seuil

- Le changement de régime est gouverné par une variable observée : Y_{t-d} la variable expliquée retardée de d périodes ou Z_t une variable exogène quelconque.
- Suivant la position de cette variable par rapport à la valeur d'un seuil λ , la dynamique de Y_t sera décrite par la première ou la deuxième équation :

$$Y_t = \begin{cases} c_1 + \beta_1 X_t + \sigma_1 \varepsilon_t & \text{si } Z_t \leq \lambda \\ c_2 + \beta_2 X_t + \sigma_2 \varepsilon_t & \text{si } Z_t > \lambda \end{cases} \quad t = 1, \dots, T \quad (5)$$

On s'intéressera à cette première classe de modèles dans le chapitre 3.

Illustration : Relation entre la consommation d'électricité et la température

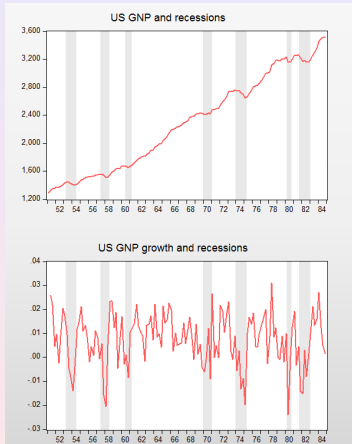


Modèle à changement de régime markovien

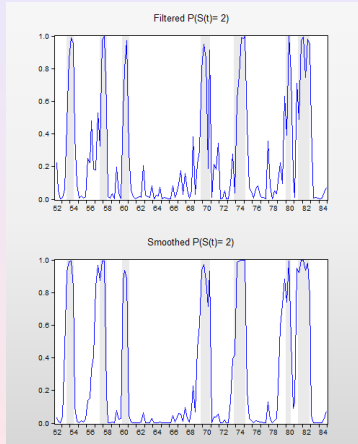
- Le changement de régime est commandé par une variable d'état *inobservable* qui suit une *chaîne de Markov* d'ordre 1.
- Ce modèle ne requiert pas de spécifier une variable de seuil et est ainsi moins sujet à des erreurs de spécification.
- La variable d'état est *inobservable* mais à l'issue de l'estimation du modèle, on est capable de calculer à chaque instant du temps la probabilité d'être dans chaque régime avec les *probabilités filtrées et lissées*.

Nous présenterons cette seconde classe de modèles dans le chapitre 4.

Illustration : datation du cycle



(a) Input



(b) Output

Arbre de régression et forêt aléatoire

- **Arbre de régression**

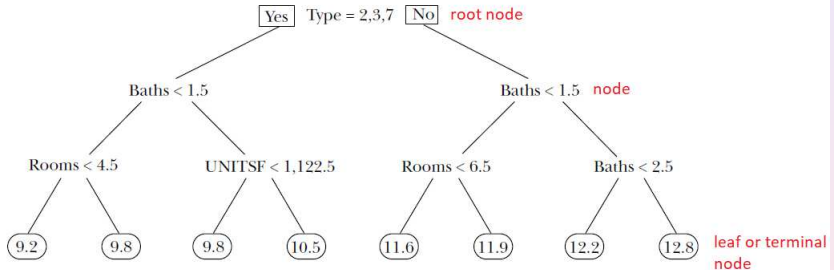
- ▶ Partition de l'espace des valeurs des variables explicatives en des régions plus petites où les interactions entre les variables sont très simples $Y = c_i$
- ▶ La partition est réalisée suivant qu'une variable explicative dépasse ou non un seuil
- ▶ Esprit similaire au modèle à seuil mais beaucoup plus de régimes et relation très simple à l'intérieur de chaque régime
- ▶ Construction de l'arbre par apprentissage sur un *training set*

- **Forêt aléatoire** : Agrégation de nombreux arbres de régression pour palier leur instabilité individuelle

On s'intéressera à cette dernière classe de modèles dans le chapitre 5.

Exemple des prix de l'immobilier

A Shallow Regression Tree Predicting House Values



Note: Based on a sample from the 2011 American Housing Survey metropolitan survey. House-value predictions are in log dollars.

Source : Mullainathan et Spiess (2017). Machine Learning : An Applied Econometric Approach. J. of Economic Perspectives.

Cadre considéré dans la suite

Pour simplifier, on limitera la présentation à :

- des modèles à 2 états, mais les spécifications présentées peuvent facilement être étendues à des représentations à plus de deux états ;
- des processus AR, c'est-à-dire avec comme variables explicatives les valeurs passées de la série, mais il est possible d'utiliser ces modèles avec des variables explicatives quelconques ;
- des processus stationnaires ou préalablement stationnarisés ($I(0)$ ou $I(0)+C$).

ECONOMETRIE NON LINEAIRE

Chapitre 2. Rappel sur le modèle autorégressif linéaire

M2 Economie - Université Paris-Dauphine

Définitions

Définition : On appelle processus *autorégressif* d'ordre p , noté $AR(p)$, un processus $(Y_t)_{t \in \mathbb{Z}}$ vérifiant :

$$Y_t = c + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

$$\Leftrightarrow \Phi(L)Y_t = c + \varepsilon_t \text{ avec } \Phi(L) = 1 - \phi_1 L - \dots - \phi_p L^p$$

où $\phi_i \in \mathbb{R}$, $\phi_p \neq 0$ et $\varepsilon_t \rightsquigarrow BB(0, \sigma_\varepsilon^2)$.

Rappel : On appelle processus *bruit blanc* un processus $(\varepsilon_t)_{t \in \mathbb{Z}}$ vérifiant :

- (i) $E(\varepsilon_t) = 0 \quad \forall t \in \mathbb{Z}$
- (ii) $V(\varepsilon_t) = \sigma_\varepsilon^2 \quad \forall t \in \mathbb{Z}$
- (iii) $cov(\varepsilon_t, \varepsilon_s) = 0 \quad \forall t, s \in \mathbb{Z}, t \neq s$

On le notera dans la suite $\varepsilon_t \rightsquigarrow BB(0, \sigma_\varepsilon^2)$.

Définitions

Définition : Soit un processus Y_t . On appelle *autocorrélation* d'ordre h de Y_t :

$$\text{cor}(Y_t, Y_{t-h}) = \frac{\text{cov}(Y_t, Y_{t-h})}{\sigma_{Y_t} \sigma_{Y_{t-h}}} \quad \forall t, h \in \mathbb{Z}$$

Définition : L'*autocorrélation partielle* d'ordre h d'un processus Y_t notée ϕ_{hh} , est définie par le dernier coefficient de projection linéaire de Y_t sur ses h plus récentes valeurs :

$$Y_t = c + \phi_{h1} Y_{t-1} + \phi_{h2} Y_{t-2} + \dots + \phi_{hh} Y_{t-h} + \varepsilon_t \quad \forall t, h \in \mathbb{Z}$$

Elle mesure la liaison entre Y_t et Y_{t-h} , l'influence des Y_{t-h+i} , $i = 1, \dots, h-1$ ayant été retirée.

Définition : On appelle *corrélogramme* simple (partiel) le graphe de $\text{cor}(Y_t, Y_{t-h})$ (de ϕ_{hh}) en fonction de h .

Propriétés

Propriété : Un processus $AR(p)$ défini par :

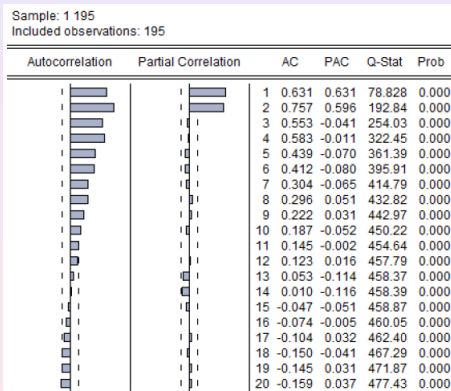
$$\Phi(L)Y_t = c + \varepsilon_t \text{ avec } \Phi(L) = 1 - \phi_1 L - \dots - \phi_p L^p$$

est stationnaire si les racines du polynôme $\Phi(L)$ sont strictement supérieures à 1 en module.

Propriété : Soit $(Y_t)_{t \in \mathbb{Z}}$ un processus $AR(p)$ stationnaire

- le corrélogramme simple de (Y_t) décroît avec l'ordre h sous la forme d'une exponentielle amortie ou une sinusoïde amortie ou la résultante des 2,
- seuls les p premiers termes du corrélogramme partiel de (Y_t) sont différents de 0.

Approche 1 : le corrélogramme



On identifie ici un processus autorégressif d'ordre 2.

Approche 2 : les critères d'information

Méthode : Estimer le processus $AR(p)$ pour plusieurs retards $p = 0, 1, \dots, p_{\max}$ et calculer pour chacun d'entre eux les critères d'information AIC et $Schwarz$. On retient le modèle pour lequel les critères sont minimaux. La définition des critères AIC et BIC est rappelée ci-dessous.

Définition : Le critère d'information de Akaike (ou AIC) est donné par :

$$AIC(K) = T \ln \hat{\sigma}_e^2 + 2K$$

et le critère de Schwarz (ou BIC) est défini par :

$$BIC(K) = T \ln \hat{\sigma}_e^2 + K \ln T$$

avec $\hat{\sigma}_e^2$ la variance résiduelle estimée, K le nombre total de paramètres du modèle et T le nombre d'observations.

Considérons pour simplifier un processus AR(1) :

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t \quad t = 2, \dots, T \quad \varepsilon_t \text{ iid } N(0, \sigma^2)$$

Le vecteur de paramètres à estimer est $\theta = (c, \phi, \sigma^2)'$

La vraisemblance **exacte** de l'échantillon (y_1, \dots, y_T) en θ s'écrit :

$$f(y_T, y_{T-1}, \dots, y_1; \theta) = f(y_1; \theta) \prod_{t=2}^T f(y_t | y_{t-1}; \theta)$$

Pour simplifier, on peut considérer simplement la vraisemblance **conditionnelle** au $p = 1$ premières observations :

$$f(y_T, \dots, y_2 | y_1; \theta) = \prod_{t=2}^T f(y_t | y_{t-1}; \theta)$$

Comme $(Y_t | Y_{t-1} = y_{t-1}) \sim N(c + \phi y_{t-1}, \sigma^2)$, on a :

$$f(y_t | y_{t-1}; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[- \frac{(y_t - c - \phi y_{t-1})^2}{2\sigma^2} \right]$$

La fonction de log-vraisemblance $L(\theta)$ s'écrit alors simplement :

$$L(\theta) = \sum_{t=2}^T \log f(y_t | y_{t-1}; \theta)$$

$$\Leftrightarrow L(\theta) = -\frac{(T-1)}{2} \log(2\pi) - \frac{(T-1)}{2} \log(\sigma^2) - \sum_{t=2}^T \frac{(y_t - c - \phi y_{t-1})^2}{2\sigma^2}$$

Il s'agira alors de chercher le vecteur θ qui maximise la log-vraisemblance.

La maximisation de $L(\theta)$ par rapport à c et ϕ revient à minimiser $\sum_{t=2}^T (y_t - c - \phi y_{t-1})^2 = (Y - X\beta)'(Y - X\beta)$.

Cela revient à considérer l'estimateur des MCO du vecteur β :

$$\hat{\beta} = (X'X)^{-1}(X'Y)$$

$$\text{avec } Y = \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_T \end{pmatrix}, X = \begin{pmatrix} 1 & y_1 \\ 1 & y_2 \\ \vdots & \vdots \\ 1 & y_{T-1} \end{pmatrix} \text{ et } \beta = \begin{pmatrix} c \\ \phi \end{pmatrix}$$

L'estimation du MV conditionnel de σ^2 est obtenu en résolvant :

$$\frac{\partial L(\theta)}{\partial \sigma^2} = \frac{-(T-1)}{2\sigma^2} + \sum_{t=2}^T \frac{(y_t - c - \phi y_{t-1})^2}{2\sigma^4} = 0$$

ce qui permet d'obtenir l'estimation de σ^2 en utilisant les estimations de c et ϕ précédentes :

$$\hat{\sigma}^2 = \sum_{t=2}^T \frac{(y_t - \hat{c} - \hat{\phi} y_{t-1})^2}{T-1} = \sum_{t=2}^T \frac{\hat{\varepsilon}_t^2}{T-1}$$

Propriétés : Les estimateurs des coefficients $\phi_i, i = 1, \dots, p$ sont convergents et asymptotiquement normalement distribués.

Conséquences :

- 1 Les intervalles de confiance de niveau $1 - \alpha$ des paramètres $\phi_i, i = 1, \dots, p$ sont donnés par :

$$\left[\hat{\phi}_i - t_{1-\alpha/2} \hat{\sigma}_{\hat{\phi}_i}, \hat{\phi}_i + t_{1-\alpha/2} \hat{\sigma}_{\hat{\phi}_i} \right]$$

avec $t_{1-\alpha/2}$ le quantile d'ordre $1 - \alpha/2$ de la loi de $St(T - K)$ avec T le nombre d'observations et K le nombre de paramètres du modèle (si $T - K > 30$, on utilisera les quantiles de la loi Normale).

- 2 Les t-statistiques usuelles peuvent être utilisées pour tester la significativité des coefficients.

Soit un processus AR(p) :

$$Y_t = c + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t \quad t = p+1, \dots, T$$

Test de significativité individuelle des coefficients

- Hypothèses : $H_0 : \phi_i = 0$ contre $H_1 : \phi_i \neq 0$
- Statistique de test : $t_{\hat{\phi}_i} = \frac{\hat{\phi}_i - 0}{\hat{\sigma}_{\hat{\phi}_i}} \underset{H_0}{\sim} St(T - K)$ avec T : le nombre d'observations et K : le nombre de paramètres estimés (ici $p + 1$)
- Remarque : $t_{\hat{\phi}_i} \underset{H_0}{\sim} N(0, 1)$ si $T - K > 30$
- Règle de décision : si $|t_{\hat{\phi}_i}| > u_{1-\alpha/2}$ (1,96 si $\alpha = 5\%$) ou $Prob < \alpha$, rejeter H_0 ; sinon, la variable Y_{t-i} peut être retirée de l'équation

Test d'autocorrélation de Ljung Box

Soit ρ_j l'autocorrélation d'ordre j du processus des erreurs ε_t .
Les **hypothèses testées** sont :

$$\begin{cases} H_0 : \rho_1 = \rho_2 = \dots = \rho_H = 0 \\ H_1 : \exists j \in \{1, 2, \dots, H\} \text{ tel que } \rho_j \neq 0 \end{cases}$$

La **statistique de Ljung-Box** est donnée par :

$$LB(H) = T(T+2) \sum_{h=1}^H \frac{\hat{\rho}_h^2}{T-h} \underset{H_0}{\sim} \chi^2(H-k)$$

avec k est le nombre de paramètres du modèle (hors constante).

Règle de décision : rejet de H_0 au seuil α si

$$LB(H) > \chi_{H-k, 1-\alpha}^2$$

Test d'homoscédasticité

On pourra également vérifier l'absence d'effets ARCH dans les résidus du modèle. On pourra appliquer à cette fin le test de Ljung Box sur les résidus estimés au carré.

Test de normalité de Jarque Bera

On pourra enfin vérifier la normalité des résidus. L'application de ce test est importante car :

- Cette hypothèse est nécessaire pour l'obtention des lois des statistiques de test et la construction des intervalles de confiance et des intervalles de prévision lorsque le nombre d'observations est petit.
- La violation de cette hypothèse peut par ailleurs indiquer la présence de points anormaux ou d'effets ARCH.

Un des tests de normalité très utilisé est le test de Jarque Bera qui consiste à comparer les moments d'ordre 3 et 4 de la série à ceux d'une loi normale.

Test de normalité de Jarque Bera (suite)

Sous H_0 , la série de résidus ε_t suit une loi normale.

La **statistique de Jarque Bera** sur une série y_t s'écrit comme suit :

$$JB = \frac{T - k}{6} \left[SK^2 + \frac{1}{4}(Kurt - 3)^2 \right] \underset{H_0}{\sim} \chi^2(2)$$

avec T le nombre d'observations, k le nombre de paramètres estimés (0 pour une série brute), $SK = \hat{\mu}_3 / \hat{\mu}_2^{3/2}$ le coefficient d'asymétrie (skewness), $Kurt = \hat{\mu}_4 / \hat{\mu}_2^2$ le coefficient d'aplatissement (kurtosis), $\hat{\mu}_k = \sum_{t=1}^T (y_t - \bar{y})^k / T$ le moment empirique centré d'ordre k .

Règle de décision : rejet de H_0 au seuil α si $JB > \chi_{2,1-\alpha}$ (5,99 si $\alpha = 5\%$).

ECONOMETRIE NON LINEAIRE

Chapitre 3. Les modèles à seuil

M2 Economie - Université Paris-Dauphine

Sommaire et références

Plan du chapitre

1. Modèle à transition brutale

- 1.1 Définitions et propriétés
- 1.2 Identification et estimation
- 1.3 Tests de linéarité
- 1.4 Tests de spécification
- 1.5 Illustrations

2. Modèle à transition lisse

- 2.1 Définitions et propriétés
- 2.2 Identification et estimation
- 2.3 Tests de linéarité
- 2.4 Tests de spécification
- 2.5 Illustrations

Ouvrages de référence

Franses P.H., van Dijk D. (2000), Non-linear time series models in empirical finance, Cambridge University Press.

Teräsvirta, T., Tjøstheim, D. et Granger, C.W.J. (2010), Modelling Nonlinear Economic Time Series, Oxford, Oxford University Press.

Principaux auteurs



(a) Howell Tong



(b) Timo Teräsvirta



(c) Bruce E. Hansen

Définition : On appelle processus SETAR d'ordres p_1 et p_2 , noté $\text{SETAR}(p_1, p_2)$, un processus $(X_t)_{t \in \mathbb{Z}}$ tel que :

$$Y_t = \begin{cases} \phi_{0,1} + \phi_{1,1} Y_{t-1} + \cdots + \phi_{p_1,1} Y_{t-p_1} + \varepsilon_t & \text{si } Y_{t-d} \leq \lambda \\ \phi_{0,2} + \phi_{1,2} Y_{t-1} + \cdots + \phi_{p_2,2} Y_{t-p_2} + \varepsilon_t & \text{si } Y_{t-d} > \lambda \end{cases}$$

ou de façon équivalente :

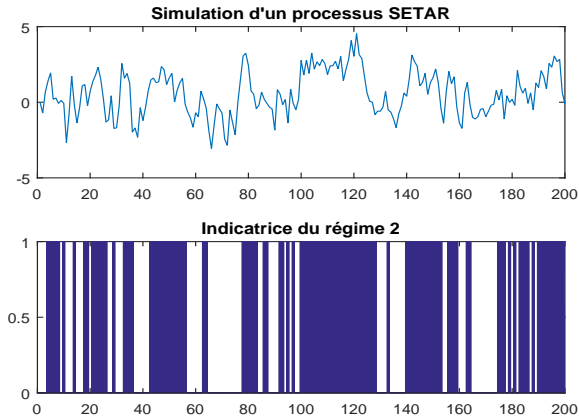
$$Y_t = (\phi_{0,1} + \phi_{1,1} Y_{t-1} + \cdots + \phi_{p_1,1} Y_{t-p_1}) \times [1 - I(Y_{t-d} > \lambda)] + (\phi_{0,2} + \phi_{1,2} Y_{t-1} + \cdots + \phi_{p_2,2} Y_{t-p_2}) \times I(Y_{t-d} > \lambda) + \varepsilon_t$$

où $I(A) = 1$ si A est vrai et 0 sinon et avec ε_t i.i.d $(0, \sigma_\varepsilon^2)$. Le terme Y_{t-d} est appelé variable de seuil, λ est le paramètre de seuil et d est le paramètre de délai.

Remarques

- 1 La variable de seuil peut être une variable exogène quelconque Z_{t-d} . On parle alors simplement de modèle TAR.
- 2 Pour simplifier les ordres p_1 et p_2 sont souvent considérés identiques dans les deux régimes, *i.e.* $p_1 = p_2 = p$.
- 3 La transition d'un régime à l'autre est brutale. Si la variable de seuil Y_{t-d} est inférieure (supérieure) à λ et cela même très légèrement, le processus est décrit par la première (seconde) équation. Dans d'autres spécifications (les modèles STAR décrits dans la section suivante), le passage du régime bas au régime haut est plus progressif.
- 4 Le choix de la variable de seuil peut être guidé par la théorie économique. On pourra également se fonder sur les résultats d'estimation (voir ci-après).

Exemple : Simulation d'un processus SETAR(1,1) de taille $T=200$ avec $d = 1$, $\lambda = 0$, $\phi_{0,1} = -0.2$, $\phi_{0,2} = 0.2$, $\phi_{1,1} = 0.5$, $\phi_{1,2} = 0.7$ et ε_t i.i.d. $N(0,1)$.



Identification des ordres p_1 et p_2

Approche 1 : Choisir $p_1 = p_2 = p$ avec p l'ordre du modèle linéaire AR(p) retenu avec **corrélogramme partiel** + **critères d'information** (voir chap.2). Approche simple mais p_1 et p_2 peuvent être différents et n'ont aucune raison de coïncider avec l'ordre p du ML.

Approche 2 : Choisir p_1 et p_2 qui minimisent les critères AIC et Schwarz du modèle SETAR ou une variante des critères AIC et Schwarz définis par Tong (1990) dans le modèle SETAR par :

$$AIC(p_1, p_2) = T_1 \ln \hat{\sigma}_1^2 + T_2 \ln \hat{\sigma}_2^2 + 2(p_1 + 1) + 2(p_2 + 1)$$

$$SC(p_1, p_2) = T_1 \ln \hat{\sigma}_1^2 + T_2 \ln \hat{\sigma}_2^2 + (p_1 + 1) \ln T_1 + (p_2 + 1) \ln T_2$$

avec T_i le nombre d'observations dans le régime i et $\hat{\sigma}_i^2$ la variance résiduelle estimée du régime i . Approche plus lourde mais estimation convergente des ordres (Sin et White, J. of Econometrics, 1996)

Estimation : Moindres Carrés Récursifs

- 1 Définir Λ l'ensemble des valeurs du seuil λ à balayer :

$$\Lambda = [a, b] \text{ avec } P(Y_{t-d} < a) = \alpha \text{ et } P(Y_{t-d} > b) = \alpha$$

On retient souvent $\alpha = 15\%$ pour avoir au minimum 15% d'observations dans chaque état.

- 2 Pour chaque seuil λ et pour un paramètre de délai d donné, estimer le modèle SETAR par les MCO :

$$Y_t = \left(\hat{\phi}_{0,1} + \hat{\phi}_{1,1} Y_{t-1} + \cdots + \hat{\phi}_{p,1} Y_{t-p} \right) \times [1 - I(Y_{t-d} > \lambda)] \\ + \left(\hat{\phi}_{0,2} + \hat{\phi}_{1,2} Y_{t-1} + \cdots + \hat{\phi}_{p,2} Y_{t-p} \right) \times I(Y_{t-d} > \lambda) + e_t(\lambda, d)$$

et calculer la variance résiduelle associée :

$$\hat{\sigma}^2(\lambda, d) = (T - p)^{-1} \sum_{t=p+1}^T e_t^2(\lambda, d)$$

Estimation : Moindres Carrés Récursifs (suite)

- ③ Retenir le seuil $\hat{\lambda}$ pour lequel la variance résiduelle est la plus petite. Les estimateurs des autres paramètres sont les estimateurs MCO $\hat{\phi}_{i,j}(\hat{\lambda}, d)$ calculés en $\hat{\lambda}$.
- ④ Répéter cette procédure pour les différentes valeurs du délai $d \in [1, d_{\max}]$.

↪ Les valeurs des paramètres pour lesquelles la variance résiduelle est la plus petite sont finalement retenues.

Estimation : Moindres Carrés Récursifs (suite)

- **Propriété** : Les estimateurs des coefficients $\phi_{i,j}$, $i = 0, \dots, p_j$ et $j = 1, 2$ sont convergents et asymptotiquement normalement distribués.
- **Conséquence** : Les intervalles de confiance de niveau $1 - \alpha$ des paramètres $\phi_{i,j}$, $i = 0, \dots, p_j$ et $j = 1, 2$ sont donnés par :

$$IC_{1-\alpha} = \left[\hat{\phi}_{i,j} - u_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{\phi}_{i,j}}, \hat{\phi}_{i,j} + u_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{\phi}_{i,j}} \right]$$

avec $u_{1-\frac{\alpha}{2}}$ le quantile d'ordre $1 - \frac{\alpha}{2}$ de la loi normale $N(0,1)$.

Remarque : Dans le cas de plus de 2 régimes, la procédure est beaucoup plus lourde : pour 3 régimes (i.e. 2 seuils), il faut estimer $d_{\max} \times N^2$ spécifications avec N le nombre de valeurs de seuils balayées. Toutefois, Bai (1997) et Bai et Perron (1998) ont proposé pour les modèles à ruptures une procédure transposable aux modèles à seuil. Dans le cas de deux seuils, on estimera d'abord le modèle à un seuil, puis pour les valeurs du seuil et du paramètre de délai obtenues $\hat{\lambda}_1$ et \hat{d} , on estimera le second seuil λ_2 dans le modèle à deux seuils. Les estimateurs des seuils et du paramètre de délai sont convergents. En outre, Bai (1997) montre que ces estimateurs peuvent être rendus asymptotiquement efficaces si la méthode est itérée au moins une fois, i.e. si pour la valeur du deuxième seuil estimée dans la deuxième étape $\hat{\lambda}_2$, on affine l'estimation du premier seuil λ_1 (voir aussi Gonzalo et Pitarakis, 2002).

Test de linéarité

- **Hypothèses** : Le jeu d'hypothèses testées est le suivant :

$$\begin{cases} H_0 : \phi_{i,1} = \phi_{i,2} & i = 0, \dots, p \\ H_1 : \exists i \in \{0, \dots, p\} / \phi_{i,1} \neq \phi_{i,2} \end{cases}$$

- **Statistique de test** : Statistique de Wald

$$W_T = \hat{\sigma}^{-2} \left(R\hat{\theta} - r \right)' \left\{ R(\mathbb{X}'\mathbb{X})^{-1} R' \right\}^{-1} \left(R\hat{\theta} - r \right)$$

avec $\hat{\sigma}^2$ la variance résiduelle estimée sous H_1 , \mathbb{X} la matrice des variables explicatives du modèle SETAR et $R\theta = r$ est la contrainte sur le vecteur des paramètres θ du modèle SETAR sous H_0 .

- **Remarque** : Utilisation alternative possible d'une statistique LR ou d'une statistique LM.

Exemple du SETAR(1,1)

$$Y_t = (c_1 + \phi_1 Y_{t-1}) \times [1 - I(Y_{t-1} > \lambda)] + (c_2 + \phi_2 Y_{t-1}) \times I(Y_{t-1} > \lambda) + \varepsilon_t \Leftrightarrow Y = \mathbb{X}\theta + \varepsilon$$

$$\text{avec } \theta = \begin{pmatrix} c_1 \\ \phi_1 \\ c_2 \\ \phi_2 \end{pmatrix}, Y = \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_T \end{pmatrix}, \mathbb{X} =$$

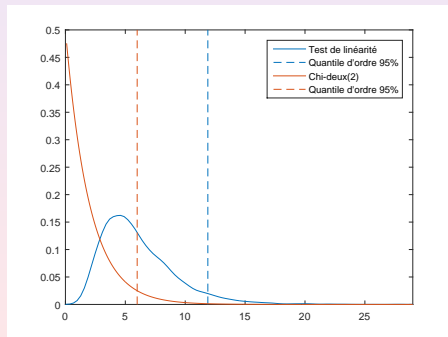
$$\begin{pmatrix} 1 \times [1 - I(y_1 > \lambda)] & y_1 [1 - I(y_1 > \lambda)] & 1 \times I(y_1 > \lambda) & y_1 I(y_1 > \lambda) \\ 1 \times [1 - I(y_2 > \lambda)] & y_2 [1 - I(y_2 > \lambda)] & 1 \times I(y_2 > \lambda) & y_2 I(y_2 > \lambda) \\ \vdots & \vdots & \vdots & \vdots \\ 1 \times [1 - I(y_{T-1} > \lambda)] & y_{T-1} [1 - I(y_{T-1} > \lambda)] & 1 \times I(y_{T-1} > \lambda) & y_{T-1} I(y_{T-1} > \lambda) \end{pmatrix}$$

La statistique de Wald s'écrit $W_T = \hat{\sigma}^{-2} (R\hat{\theta} - r)' \{ R(\mathbb{X}'\mathbb{X})^{-1} R' \}^{-1} (R\hat{\theta} - r)$

$$\text{avec } R = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \text{ et } r = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- **Problème** : Compte tenu de l'existence de paramètres de nuisance non identifiés sous H_0 (d et λ), W_T ne suit pas une distribution du Chi-deux (sous H_0 , une variation des paramètres d et λ laisse inchangée la vraisemblance. Davies (1987), Andrews et Ploberger (1994) ou Hansen (1996) montrent alors que la statistique du test de linéarité ne suit pas une loi usuelle dans ce cas).
- **Conséquence** : La distribution doit alors être obtenue par simulations de Monte Carlo ou par bootstrap (Hansen, 1996) et dépend des moments des régresseurs spécifiques aux observations utilisées, ce qui exclut une tabulation générale des valeurs critiques.

- **Illustration** Loi du $\chi^2(2)$ et distribution de la statistique de linéarité d'un SETAR(1,1) obtenue par simulations de Monte Carlo (10000 tirages). Quantile d'ordre 95% = 11.88 contre 5.99 dans le cas de la loi du $\chi^2(2)$. L'utilisation des lois usuelles est donc fortement déconseillée !



Les valeurs critiques étant inconnues, nous utiliserons la procédure de Hansen (1996) afin d'en obtenir une approximation. Cette méthode comporte trois étapes.

- 1 Simuler T observations à partir du modèle estimé sous H_0 :

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t \quad t = 1, \dots, T$$

On pourra utiliser, comme valeurs initiales y_1, \dots, y_p , les p premières observations de la chronique.

- 2 A partir des observations simulées, on estime ensuite le modèle sous l'hypothèse alternative :

$$Y_t = \begin{cases} \phi_{0,1} + \phi_{1,1} Y_{t-1} + \dots + \phi_{p,1} Y_{t-p} + \varepsilon_t & \text{si } Y_{t-d} \leq \lambda \\ \phi_{0,2} + \phi_{1,2} Y_{t-1} + \dots + \phi_{p,2} Y_{t-p} + \varepsilon_t & \text{si } Y_{t-d} > \lambda \end{cases}$$

③ On calcule enfin la statistique de test de linéarité W_T^S .

Cette procédure est répliquée un grand nombre de fois. La distribution empirique peut être établie à partir de la séquence de statistiques de test calculées. La p-value du test p_t peut également être obtenue comme le pourcentage de statistiques simulées W_T^S supérieures à la statistique W_T^* calculée sur l'échantillon observé : $p_t = Pr \{ W_T^S > W_T^* \}$.

Remarque : dans l'étape 1, on simulera des résidus ε_t i.i.d $N(0, \sigma_e^2)$ avec σ_e^2 la variance estimée des résidus du modèle linéaire estimé (*randn* sous matlab) ou on tirera au sort dans les résidus estimés du modèle linéaire (*bootstrap*). Cette dernière approche a l'avantage de ne pas nécessiter d'hypothèses sur la distribution du bruit.

Tests de spécification

Nous appliquerons à ce niveau les tests de significativité individuelle des coefficients du modèle SETAR.

Nous appliquerons également les tests usuels aux résidus estimés du modèle SETAR :

- Test de Ljung Box pour vérifier que les résidus sont non autocorrélés
- Test d'effets ARCH, test de White pour vérifier que les résidus sont non homoscédastiques
- Test de Jarque Bera pour vérifier la normalité des résidus

- **Définition** : On appelle processus STAR d'ordres p_1 et p_2 , noté $\text{STAR}(p_1, p_2)$, un processus $(Y_t)_{t \in \mathbb{Z}}$:

$$Y_t = (\phi_{0,1} + \phi_{1,1} Y_{t-1} + \dots + \phi_{p_1,1} Y_{t-p_1}) \times [1 - F(Y_{t-d}; \gamma, \lambda)] \\ + (\phi_{0,2} + \phi_{1,2} Y_{t-1} + \dots + \phi_{p_2,2} Y_{t-p_2}) \times F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t$$

avec F une fonction de transition continue bornée par 0 et 1 et Y_{t-d} la variable de transition.

- **Remarque** : Pour simplifier, on considérera souvent que $p_1 = p_2 = p$.
- **Notation synthétique** :

$$Y_t = \phi'_1 \chi_{t-1,1} \{1 - F(Y_{t-d}; \gamma, \lambda)\} + \phi'_2 \chi_{t-1,2} F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t$$

avec $\phi'_j = (\phi_{0,j}, \phi_{1,j}, \dots, \phi_{p_j,j})$, $\chi_{t-1,j} = (1, Y_{t-1}, \dots, Y_{t-p_j})'$.

- **Deux définitions alternatives de la fonction F :**

- ▷ la fonction de transition logistique (LSTAR) :

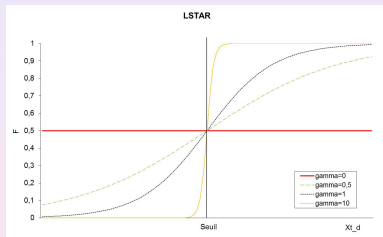
$$F(Y_{t-d}; \gamma, \lambda) = \{1 + \exp[-\gamma(Y_{t-d} - \lambda)]\}^{-1}$$

- ▷ la fonction de transition exponentielle (ESTAR) :

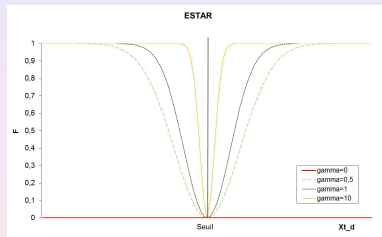
$$F(Y_{t-d}; \gamma, \lambda) = 1 - \exp\left[-\gamma(Y_{t-d} - \lambda)^2\right]$$

λ est le paramètre de seuil et γ est un paramètre strictement positif mesurant la vitesse de transition entre les deux régimes. Plus γ est grand et plus la transition entre les deux régimes est brutale (voir figure ci-après).

Illustration : Fonctions de transition du modèle STAR



(a) transition logistique



(b) transition exponentielle

Cas particuliers :

si $\gamma = 0$, modèle linéaire

si $\gamma \rightarrow \infty$ dans le modèle LSTAR, modèle TAR

● La nature des changements de régime

1) Des changements graduels : Ces deux modèles impliquent l'existence de deux régimes extrêmes distincts dans la dynamique des séries correspondant aux valeurs extrêmes de la fonction de transition 0 et 1 et une continuité de régimes associés aux valeurs intermédiaires de la fonction de transition.

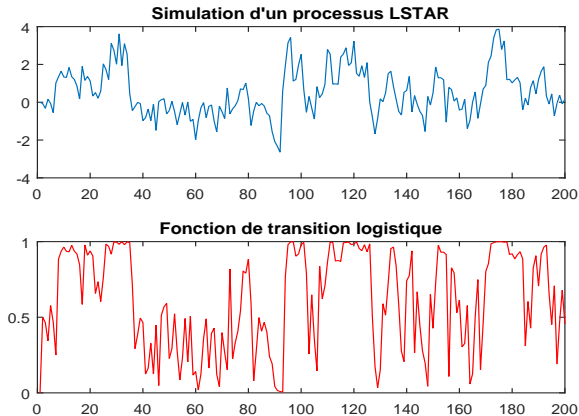
2) Des changements différents dans les LSTAR et ESTAR :

- ▷ Fonction logistique en forme de 'S' \Rightarrow deux régimes extrêmes associés aux valeurs hautes et basses de Y_{t-d} relativement au seuil λ .
- ▷ Fonction exponentielle en forme de 'U' \Rightarrow un régime central lorsque Y_{t-d} est proche de λ et deux régimes extrêmes de même dynamique.

Remarques

- 1 Si $\gamma = 0$, la fonction de transition est constante (et vaut 0.5 dans la spécification LSTAR et 0 dans la spécification ESTAR) ; le modèle STAR(p) se ramène alors un modèle linéaire usuel AR(p). Le modèle AR est donc un cas particulier des modèles STAR (LSTAR comme ESTAR).
- 2 Si $\gamma \rightarrow \infty$, la fonction de transition logistique tend vers la fonction indicatrice utilisée dans les modèles TAR (puisque si $\gamma \rightarrow \infty$, $F(Y_{t-d}) \rightarrow 1$ si $Y_{t-d} > \lambda$ et $F(Y_{t-d}) \rightarrow 0$ sinon). Le modèle TAR est un cas particulier du modèle LSTAR.

Exemple : Simulation d'un processus LSTAR(1,1) de taille $T=200$ avec $d = 1$, $\lambda = 0$, $\gamma = 2$, $\phi_{0,1} = -0.2$, $\phi_{0,2} = 0.2$, $\phi_{1,1} = 0.5$, $\phi_{1,2} = 0.7$ et ε_t i.i.d. $N(0,1)$.



Identification des ordres p_1 et p_2

Les deux approches présentées pour les modèles TAR sont également applicables aux modèles STAR. On utilisera de façon alternative :

- l'ordre du processus autorégressif linéaire obtenu dans le modèle linéaire (corrélogramme ou critères d'information)
- les critères d'information AIC et Schwarz qu'il s'agira de minimiser dans le modèle non linéaire.

Estimation : MC non linéaires

Méthode : Les différents coefficients du modèle STAR sont estimés par les moindres carrés non-linéaires. Les paramètres $\theta = (\phi'_1, \phi'_2, \gamma, \lambda)$ sont solution de :

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^T [Y_t - G(X_t; \theta)]^2$$

où $G(X_t; \theta) = \phi'_1 \chi_{t-1} \{1 - F(Y_{t-d}; \gamma, \lambda)\} + \phi'_2 \chi_{t-1} F(Y_{t-d}; \gamma, \lambda)$,
 $\chi_{t-1} = (1, Y_{t-1}, \dots, Y_{t-p})'$ et $\phi'_j = (\phi_{0,j}, \phi_{1,j}, \dots, \phi_{p,j})$, $j = 1, 2$.

Le support du paramètre γ n'étant pas fini, on n'estime pas de façon récursive tous les modèles possibles comme pour le SETAR mais on utilise d'un algorithme d'optimisation numérique (par ex. *Newton-Raphson*).

Remarques

- 1 Pour faciliter la convergence de l'algorithme d'optimisation numérique, il est conseillé de diviser γ par l'écart-type de la série et d'utiliser une fonction de transition de la forme :

▷ $F(Y_{t-d}) = \{1 + \exp[-\gamma(Y_{t-d} - \lambda)/\sigma_y]\}^{-1}$ si LSTAR

▷ $F(Y_{t-d}) = 1 - \exp[-\gamma(Y_{t-d} - c)^2/\sigma_y^2]$ si modèle ESTAR.

On pourra alors choisir 1 comme valeur initiale pour γ (et les valeurs estimées du ML pour les autres coefficients).

- 2 En pratique, il s'avère difficile d'obtenir une estimation précise du paramètre de lissage γ . L'estimateur de γ est généralement imprécis et souvent non significatif au vu de sa t-statistique. En conséquence, la non significativité de ce paramètre ne doit pas être interprétée comme une preuve de linéarité du processus.

- **Propriété** : Sous certaines conditions de régularité, les estimateurs des coefficients $\phi_{i,j}$, $i = 0, \dots, p$ et $j = 1, 2$ sont convergents et asymptotiquement normalement distribués.
- **Conséquence** : De même, les intervalles de confiance de niveau $1 - \alpha$ des paramètres $\phi_{i,j}$, $i = 0, \dots, p$ et $j = 1, 2$ sont donnés par :

$$IC_{1-\alpha} = \left[\hat{\phi}_{i,j} - u_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{\phi}_{i,j}}, \hat{\phi}_{i,j} + u_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{\phi}_{i,j}} \right]$$

avec $u_{1-\frac{\alpha}{2}}$ le quantile d'ordre $1 - \frac{\alpha}{2}$ de la loi normale $N(0,1)$.

- L'**hypothèse nulle** testée admet ici deux écritures possibles :

$$H_0 : \phi_{i,1} = \phi_{i,2} \quad i = 0, \dots, p \Leftrightarrow H_0 : \gamma = 0$$

Dans les 2 cas, l'équation décrivant la dynamique de la variable ne varie pas dans le temps. La construction du test de linéarité présentée dans la suite repose sur la deuxième formulation de H_0 .

- Dans le cas des modèles STAR, le **problème des paramètres de nuisance** (γ , λ et d) non identifiés sous H_0 se pose à nouveau. Toutefois, on peut se ramener à un cadre standard en utilisant un développement de Taylor des fonctions de transition. Dans l'équation auxiliaire obtenue, le problème de non identification ne se pose plus et les lois usuelles peuvent être utilisées.

Rappel : Le développement de Taylor à l'ordre 1 d'une fonction à n variables

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et supposons que les dérivées partielles premières $f'_{x_i}(x_0)$ sont définies et continues $\forall i$.

Alors le développement de Taylor à l'ordre 1 (DT(1) dans la suite) de f en x_0 s'écrit :

$$f(x) = f(x_0) + \sum_{i=1}^n f'_{x_i}(x_0)(x_i - x_{i0}) + R \text{ avec } R \xrightarrow{x \rightarrow x_0} 0$$

Reformulation du modèle LSTAR et test de linéarité

- Reformulation du modèle LSTAR(p,p) avec un DT(1) de la fonction de transition logistique :

$$Y_t = \beta_0 + \beta_1' \tilde{x}_t + \beta_2' \tilde{x}_t y_{t-d} + \varepsilon_t \quad (a)$$

avec $\tilde{x}_t = (y_{t-1}, \dots, y_{t-p})'$. On montre que $\beta_2 = 0$ si $\gamma = 0$.

- L'hypothèse nulle du test de linéarité devient donc :

$$H_0 : \gamma = 0 \Leftrightarrow \beta_2' = (0, \dots, 0)$$

- Statistique de test et loi sous H_0 :

$$LM = T \left(\frac{SCR_0 - SCR_1}{SCR_0} \right) \underset{H_0}{\sim} \chi^2(p)$$

avec SCR_0 la SCR du modèle linéaire et SCR_1 la SCR de la régression auxiliaire.

Démonstration

Soit un modèle LSTAR(p,p) :

$$Y_t = (\phi_{0,1} + \phi_{1,1} Y_{t-1} + \dots + \phi_{p,1} Y_{t-p}) \times [1 - F(Y_{t-d}; \gamma, \lambda)] \\ + (\phi_{0,2} + \phi_{1,2} Y_{t-1} + \dots + \phi_{p,2} Y_{t-p}) \times F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t$$

avec $F(Y_{t-d}; \gamma, \lambda) = \{1 + \exp[-\gamma (Y_{t-d} - \lambda)]\}^{-1}$.

Le modèle LSTAR(p,p) peut se réécrire comme suit :

$$Y_t = (\phi_{0,1} + \phi_{1,1} Y_{t-1} + \dots + \phi_{p,1} Y_{t-p_1}) \times [1 - F(Y_{t-d}; \gamma, \lambda)] \\ + (\phi_{0,2} + \phi_{1,2} Y_{t-1} + \dots + \phi_{p,2} Y_{t-p_1}) \times F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t$$

$$\iff Y_t = \phi'_1 \chi_{t-1} + (\phi'_2 - \phi'_1) \chi_{t-1} F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t$$

$$\iff Y_t = \frac{1}{2} (\phi'_1 + \phi'_2) \chi_{t-1} + (\phi'_2 - \phi'_1) \chi_{t-1} F^*(Y_{t-d}; \gamma, \lambda) + \varepsilon_t^*$$

Démonstration (suite)

avec $\phi'_j = (\phi_{0,j}, \phi_{1,j}, \dots, \phi_{p,j})$, $\chi_{t-1} = (1, Y_{t-1}, \dots, Y_{t-p})$ et $F^*(Y_{t-d}; \gamma, \lambda) = F(Y_{t-d}; \gamma, \lambda) - 1/2$. On notera que si $\gamma = 0.5$, $F = 0.5$ d'où $F^* = 0$.

Le développement de Taylor à l'ordre 1 de la fonction F^* au voisinage de $\gamma = 0$ permet de réécrire F^* comme suit :

$$F^*(Y_{t-d}; \gamma, \lambda) \simeq F^*(Y_{t-d}; 0, \lambda) + (\gamma - 0) \left(\frac{\partial F^*(Y_{t-d}; \gamma, \lambda)}{\partial \gamma} \Big|_{\gamma=0} \right)$$

$$\text{Or } F^*(Y_{t-d}; 0, \lambda) = 0$$

$$\text{et } \frac{\partial F^*(Y_{t-d}; \gamma, \lambda)}{\partial \gamma} \Big|_{\gamma=0} = \frac{(Y_{t-d} - \lambda) \exp[-\gamma (Y_{t-d} - \lambda)]}{\{1 + \exp[-\gamma (Y_{t-d} - \lambda)]\}^2} \Big|_{\gamma=0} = \frac{Y_{t-d} - \lambda}{4}, \text{ d'où :}$$

$$F^*(Y_{t-d}; \gamma, \lambda) \simeq \frac{1}{4} \gamma (Y_{t-d} - \lambda) **$$

Démonstration (suite)

En utilisant les relations (*) et (**), on obtient alors :

$$\begin{aligned} Y_t &= \frac{1}{2} (\phi'_1 + \phi'_2) \chi_{t-1} + (\phi'_2 - \phi'_1) \chi_{t-1} \times \frac{1}{4} \gamma (Y_{t-d} - \lambda) + \varepsilon_t \\ &= \frac{1}{2} (\phi_{0,1} + \phi_{0,2}) - (\phi_{0,2} - \phi_{0,1}) \times \frac{1}{4} \gamma \lambda + (\phi_{0,2} - \phi_{0,1}) \times \frac{1}{4} \gamma Y_{t-d} \\ &+ \frac{1}{2} \sum_{i=1}^p (\phi_{i,1} + \phi_{i,2}) Y_{t-i} - \frac{1}{4} \gamma \lambda \sum_{i=1}^p (\phi_{i,2} - \phi_{i,1}) Y_{t-i} \\ &+ \frac{1}{4} \gamma \sum_{i=1}^p (\phi_{i,2} - \phi_{i,1}) Y_{t-i} Y_{t-d} + \varepsilon_t \end{aligned}$$

$$\begin{aligned} Y_t &= \frac{1}{2} (\phi_{0,1} + \phi_{0,2}) - (\phi_{0,2} - \phi_{0,1}) \times \frac{1}{4} \gamma \lambda + (\phi_{0,2} - \phi_{0,1}) \times \frac{1}{4} \gamma Y_{t-d} \\ &+ \sum_{i=1}^p \left[\frac{1}{2} (\phi_{i,1} + \phi_{i,2}) - \frac{1}{4} \gamma \lambda (\phi_{i,2} - \phi_{i,1}) \right] Y_{t-i} \\ &+ \frac{1}{4} \gamma \sum_{i=1}^p (\phi_{i,2} - \phi_{i,1}) Y_{t-i} Y_{t-d} + \varepsilon_t \end{aligned}$$

Démonstration (suite)

soit $Y_t = \beta_0 + \beta_1' \tilde{x}_t + \beta_2' \tilde{x}_t Y_{t-d} + \varepsilon_t$ avec $\tilde{x}_t = (y_{t-1}, \dots, y_{t-p})'$ et des coefficients β_i tels que :

$$\beta_0 = \frac{1}{2} (\phi_{0,1} + \phi_{0,2}) - \frac{1}{4} \gamma \lambda (\phi_{0,2} - \phi_{0,1})$$

$$\beta_1 = (\beta_{1,1}, \dots, \beta_{1,p})' \text{ avec}$$

$$\begin{aligned} \beta_{1,i} &= \frac{1}{2} (\phi_{i,1} + \phi_{i,2}) - \frac{1}{4} \gamma \lambda (\phi_{i,2} - \phi_{i,1}) \text{ si } i \neq d \\ \beta_{1,d} &= \frac{1}{2} (\phi_{d,1} + \phi_{d,2}) - \frac{1}{4} \gamma \lambda (\phi_{d,2} - \phi_{d,1}) + \frac{1}{4} \gamma (\phi_{0,2} - \phi_{0,1}) \end{aligned}$$

$$\beta_2 = (\beta_{2,1}, \dots, \beta_{2,p})' \text{ avec } \beta_{2,i} = \frac{1}{4} \gamma (\phi_{i,2} - \phi_{i,1})$$

Sous $H_0 : \gamma = 0, \beta_2' = (0, \dots, 0)$.

Remarque : Le test fondé sur un développement de Taylor à l'ordre 1 ne sera pas puissant si seule la constante diffère entre les régimes (β_2 est construit à partir des seuls coefficients $\phi_{i,j}$, $i = 1, \dots, p$). Pour palier ce problème, il est possible d'utiliser un développement de Taylor à l'ordre 3 de la fonction de transition, ce qui conduit à la régression auxiliaire suivante :

$$Y_t = \beta_0 + \beta'_1 \tilde{x}_t + \beta'_2 \tilde{x}_t y_{t-d} + \beta'_3 \tilde{x}_t y_{t-d}^2 + \beta'_4 \tilde{x}_t y_{t-d}^3 + \varepsilon_t$$

On montre dans ce cadre que le test de $\gamma = 0$ est équivalent au test des contraintes $\beta_2 = \beta_3 = \beta_4 = 0$. Comme les coefficients β_2 et β_3 dépendent des constantes, le test sera donc puissant en cas d'une variation de la constante seulement entre les régimes. Sous H_0 , la statistique LM de l'hypothèse de linéarité suit une distribution $\chi^2(3p)$.

Reformulation du modèle ESTAR et test de linéarité

- Reformulation du modèle LSTAR(p,p) avec un DT(1) de la fonction de transition exponentielle :

$$Y_t = \beta_0 + \beta'_1 \tilde{x}_t + \beta'_2 \tilde{x}_t y_{t-d} + \beta'_3 \tilde{x}_t y_{t-d}^2 + \varepsilon_t \quad (b)$$

avec $\tilde{x}_t = (y_{t-1}, \dots, y_{t-p})$. On montre que $\beta_2 = \beta_3 = 0$ si $\gamma = 0$.

- L'hypothèse nulle du test de linéarité devient donc :

$$H_0 : \gamma = 0 \Leftrightarrow \beta'_2 = \beta'_3 = (0, \dots, 0)$$

- Statistique de test et loi sous H_0 :

$$LM = T \left(\frac{SCR_0 - SCR_1}{SCR_0} \right) \underset{H_0}{\sim} \chi^2(2p)$$

avec SCR_0 la SCR du modèle linéaire et SCR_1 la SCR de la régression auxiliaire.

Mise en œuvre du test de linéarité

En pratique, on procède en trois temps pour tester un modèle STAR contre un modèle AR linéaire.

- 1 Estimation du modèle linéaire, calcul des résidus \hat{u}_t et la SCR associée à H_0 : $SCR_0 = \sum_{t=1}^T \hat{u}_t^2$
- 2 Estimation de l'équation auxiliaire (a) ou (b) dont on déduit les résidus \hat{e} et la SCR sous H_1 $SCR_1 = \sum_{t=1}^T \hat{e}_t^2$
- 3 Calcul de la statistique LM :

$$LM = T \left(\frac{SCR_0 - SCR_1}{SCR_0} \right) \underset{H_0}{\sim} \chi^2(\nu)$$

avec $\nu = p$ dans la spécification LSTAR et $\nu = 2p$ dans le modèle ESTAR.

Remarque : Dans de petits échantillons, la taille empirique (fréquence de rejet à tort de H_0) de la statistique LM excède la taille nominale (seuil α). On recommande alors l'utilisation de la statistique de Fisher :

$$F_{LM} = \frac{(SCR_0 - SCR_1) / \nu}{SCR_1 / (T - K)} \underset{H_0}{\sim} F(\nu, T - K)$$

avec K le nombre de variables explicatives dans la régression auxiliaire, $\nu = p$ dans la spécification LSTAR et $\nu = 2p$ dans le modèle ESTAR.

Tests de spécification (Eitrheim et Teräsvirta, 1996)

Présentation générale : Considérons tout d'abord un modèle général non-linéaire autorégressif d'ordre p dont les résidus sont éventuellement autocorrélés :

$$y_t = G(w_t; \Psi) + u_t \quad t = 1, \dots, T \quad (\text{eq_spec})$$

$$u_t = a_1 u_{t-1} + \dots + a_q u_{t-q} + \varepsilon_t = a' v_t + \varepsilon_t$$

avec $w_t = (1, y_{t-1}, \dots, y_{t-p})'$, $\Psi = (\Psi_1, \dots, \Psi_r)$ et $a_q \neq 0$.

L'hypothèse nulle de non autocorrélation des résidus est :

$$H_0 : a_1 = \dots = a_q = 0$$

Tests de spécification (suite)

Le test peut être mené suivant les quatre étapes suivantes :

- 1 Estimer le modèle (eq_spec) et en récupérer les résidus \hat{u}_t .
- 2 Régresser \hat{u}_t sur les dérivées $\partial G(w_t; \hat{\Psi}) / \partial \Psi$ avec $\hat{\Psi}$ l'estimateur des moindres carrés non-linéaires de Ψ . Récupérer les résidus de la régression \tilde{u}_t et calculer la somme des carrés des résidus SCR_0 .
- 3 Régresser \tilde{u}_t sur \hat{v}_t et $\partial G(w_t; \hat{\Psi}) / \partial \Psi$. Récupérer les résidus et calculer la somme des carrés des résidus SCR_1 .
- 4 Calculer la statistique du multiplicateur de Lagrange

$$LM = T \frac{SCR_0 - SCR_1}{SCR_0} \underset{H_0}{\sim} \chi^2(q)$$

avec T le nombre d'observations et rejeter H_0 si

$$LM > \chi_{q, 1-\alpha}.$$

Tests de spécification (suite)

Pour de petits échantillons, il est préférable d'utiliser la statistique de Fisher :

$$F_{LM} = \frac{(SCR_0 - SCR_1) / q}{SCR_1 / (T - n - q)} \underset{H_0}{\sim} F(q, T - n - q)$$

avec T le nombre d'observations, n la dimension du vecteur $\partial G(w_t; \hat{\Psi}) / \partial \Psi$ et q l'ordre d'autocorrélation testé.

Tests de spécification (suite)

Appliquons cette méthodologie générale au modèle STAR présenté précédemment :

$$\begin{aligned}
 Y_t &= \phi'_1 \chi_{t-1} \times [1 - F(Y_{t-d}; \gamma, \lambda)] + \phi'_2 \chi_{t-1} \times F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t \\
 &= \phi'_1 \chi_{t-1} + (\phi'_2 - \phi'_1) \chi_{t-1} F(Y_{t-d}; \gamma, \lambda) + \varepsilon_t
 \end{aligned}$$

avec $\phi'_j = (\phi_{0,j}, \phi_{1,j}, \dots, \phi_{p,j})$ et $\chi_{t-1} = (1, Y_{t-1}, \dots, Y_{t-p})'$.

Tests de spécification (suite)

On appliquera donc les quatre étapes précédentes avec le vecteur $\partial G(w_t; \hat{\Psi}) / \partial \Psi$ composé des dérivées suivantes :

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \phi_{0,1}} = 1 - F(Y_{t-d}; \hat{\gamma}, \hat{\lambda}) \quad \frac{\partial G(w_t; \hat{\Psi})}{\partial \phi_{0,2}} = F(Y_{t-d}; \hat{\gamma}, \hat{\lambda})$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \phi_{i,1}} = y_{t-i} \left\{ 1 - F(Y_{t-d}; \hat{\gamma}, \hat{\lambda}) \right\} \quad i = 1, \dots, p$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \phi_{i,2}} = y_{t-i} F(Y_{t-d}; \hat{\gamma}, \hat{\lambda}) \quad i = 1, \dots, p$$

Tests de spécification (suite)

et dans le cas du modèle LSTAR où

$$F(Y_{t-d}; \gamma, \lambda = \{1 + \exp[-\gamma (Y_{t-d} - \lambda)]\}^{-1} :$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \lambda} = - \frac{\hat{\gamma} \exp[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})]}{\left\{1 + \exp[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})]\right\}^2} (\hat{\phi}'_2 - \hat{\phi}'_1) \chi_{t-1}$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \gamma} = - \frac{(Y_{t-d} - \hat{\lambda}) \exp[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})]}{\left\{1 + \exp[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})]\right\}^2} (\hat{\phi}'_2 - \hat{\phi}'_1) \chi_{t-1}$$

Tests de spécification (suite)

et dans le cas du modèle ESTAR où

$$F(Y_{t-d}; \gamma, \lambda = \left\{ 1 - \exp \left[-\gamma (Y_{t-d} - \lambda)^2 \right] \right\} :$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \lambda} = -2\hat{\gamma} (Y_{t-d} - \hat{\lambda}) \exp \left[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})^2 \right] (\hat{\phi}'_2 - \hat{\phi}'_1) \chi_{t-1}$$

$$\frac{\partial G(w_t; \hat{\Psi})}{\partial \gamma} = (Y_{t-d} - \hat{\lambda})^2 \exp \left[-\hat{\gamma} (Y_{t-d} - \hat{\lambda})^2 \right] (\hat{\phi}'_2 - \hat{\phi}'_1) \chi_{t-1}$$

- Cycle des affaires (Teräsvirta & Anderson, JASA 1992, Potter, JAE 1995, Pesaran & Potter, JEDC 1997, Ben Salem & Perraudin, 2001, Galvao, IJF 2013, Egert J. Macro 2015)
- Emploi et chômage (Perraudin, 1995, Rothman, Rev Eco Stat 1998, Koop et Potter, JBES 1999)
- Taux de change (Taylor, Econometrica 2001, Bec, Ben Salem et Carrasco, JBES 2004)
- Taux d'intérêt (Gospodinov, J. Fin. Econometrics, 2005, Delatte, Fouquau, Portes, R. of Finance 2017)
- Rendements boursiers (Griffin, Nardari et Stulz, R. financial studies 2007, Brana et Prat, Eco modelling 2016)
- Conso d'élec et température (Moral-Carcedo, Vicens-Otero, Bessec et Fouquau, Blasquez et al., Energy Eco 2005, 08, 13)

Algorithme d'optimisation numérique : Newton Raphson

Soit une fonction $f(\theta)$, $\theta = (\theta_1, \dots, \theta_n)' \in \mathbb{R}^n$ dont on cherche le minimum θ^* .

Lorsqu'il n'est pas possible d'obtenir des expressions analytiques des solutions des conditions du premier ordre (CPO), le minimum ne peut être trouvé par une méthode itérative.

Le principe est le suivant : on part de valeurs initiales des paramètres et l'on affine à chaque étape une approximation du vecteur solution des CPO fondée sur un développement de Taylor et cela jusqu'à ce que les paramètres ne varient plus.

Rappel : Soit g une fonction. Le développement de Taylor à l'ordre 1 de g en x_0 s'écrit :

$$g(x) = g(x_0) + \left. \frac{dg}{dx} \right|_{x=x_0} (x - x_0) + R \text{ avec } R \xrightarrow{x \rightarrow x_0} 0$$

Appliquons cette approximation au vecteur gradient On écrit tout d'abord les CPO suivantes que doit satisfaire le vecteur solution θ^* dès lors que la solution est intérieure et que la fonction f est dérivable en tout point :

$$\theta^* / \frac{df}{d\theta} (\theta^*) = 0 \quad (\text{A0})$$

En procédant à un développement de Taylor à l'ordre 1 du gradient $\frac{df}{d\theta}$ au voisinage de θ^* , on obtient :

$$\frac{df}{d\theta} (\theta^*) \approx \frac{df}{d\theta} (\theta_0) + \frac{d^2f}{d\theta^2} (\theta_0) (\theta^* - \theta_0) \quad (\text{A1})$$

où $\frac{df}{d\theta}(\theta_0)$ et $\frac{d^2f}{d\theta^2}(\theta_0)$ sont définis comme suit :

$$\frac{df}{d\theta}(\theta_0) = \begin{pmatrix} \frac{\partial f}{\partial \theta_1}(\theta_0) \\ \frac{\partial f}{\partial \theta_2}(\theta_0) \\ \vdots \\ \frac{\partial f}{\partial \theta_n}(\theta_0) \end{pmatrix}$$

$$\frac{d^2f}{d\theta^2}(\theta_0) = \begin{pmatrix} \frac{\partial^2 f}{\partial \theta_1^2}(\theta_0) & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_2}(\theta_0) & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_n}(\theta_0) \\ \frac{\partial^2 f}{\partial \theta_1 \partial \theta_2}(\theta_0) & \frac{\partial^2 f}{\partial \theta_2^2}(\theta_0) & \cdots & \frac{\partial^2 f}{\partial \theta_2 \partial \theta_n}(\theta_0) \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial \theta_1 \partial \theta_n}(\theta_0) & \frac{\partial^2 f}{\partial \theta_2 \partial \theta_n}(\theta_0) & \cdots & \frac{\partial^2 f}{\partial \theta_n^2}(\theta_0) \end{pmatrix}$$

En combinant les égalités (A0) et (A1), on obtient :

$$\underbrace{\frac{df}{d\theta}(\theta^*)}_0 \approx \frac{df}{d\theta}(\theta_0) + \frac{d^2f}{d\theta^2}(\theta_0)(\theta^* - \theta_0)$$

soit en isolant θ^* :

$$\theta^* = \theta_0 - \left[\frac{d^2f}{d\theta^2}(\theta_0) \right]^{-1} \frac{df}{d\theta}(\theta_0)$$

Il s'agira alors au cours d'itérations successives d'améliorer cette approximation. Formellement, en notant k , la k^e itération, on a :

$$\theta_{k+1} = \theta_k - \left[\frac{d^2f}{d\theta^2}(\theta_k) \right]^{-1} \frac{df}{d\theta}(\theta_k)$$

Lorsque la variation des paramètres entre deux itérations successives ou de façon équivalente lorsque les gradients sont inférieurs à un critère de convergence spécifié, le minimum θ^* est supposé obtenu.

Remarques

- 1 rien n'assure que la solution obtenue est un minimum global et non local. Tout dépend du choix des conditions initiales. Il est donc fortement recommandé en pratique de relancer la procédure d'optimisation pour plusieurs jeux de paramètres initiaux.
- 2 Si on ne dispose pas des expressions analytiques du gradient et du Hessien, on peut procéder à des approximations numériques de leurs éléments.

ECONOMETRIE NON LINEAIRE

Chapitre 4. Les modèles à changement de régime markovien

M2 Economie - Université Paris-Dauphine

Sommaire et références

Plan du chapitre

1. Définitions et propriétés
2. Identification et estimation
3. Tests de linéarité
4. Tests de spécification
5. Extensions
6. Illustrations

Ouvrages de référence

Franses P.H., van Dijk D., Non-linear time series models in empirical finance, Cambridge University Press, 2000.

Hamilton, J. D., Time Series Analysis, Princeton : Princeton University Press, 1994.

Principal auteur : James D. Hamilton



- Professeur à l'University of California à San Diego

- Recherche en économétrie des séries temporelles et économie de l'énergie
- Ouvrages *Time Series Analysis* et *Advances in Markov-Switching models*
- Article majeur : A new approach to the economic analysis of nonstationary time series and the business cycle, *Econometrica*, 1989

Définition

- **Définition** : Soit S_t une variable aléatoire prenant des valeurs entières $\{1, 2\}$ caractérisant l'état d'un système. On dira que S_t suit une *chaîne de Markov* d'ordre 1 si :

$$P(S_t = i | S_{t-1} = j) = P(S_t = i | S_{t-1} = j, S_{t-2} = k, \dots, I_{t-1})$$

avec $i, j, k = \{1, 2\}$ et I_{t-1} l'ensemble des variables exogènes connues à la date $t - 1$.

- **Définition** : On dira par ailleurs que la chaîne de Markov S_t est *homogène* (ou stationnaire) si :

$$P(S_t = i | S_{t-1} = j) = \dots = P(S_2 = i | S_1 = j) = p_{ji} \quad \forall t$$

Dans le cas de deux états, la dynamique de S_t est alors définie par les 4 probabilités suivantes : $p_{ji}, i, j, = \{1, 2\}$. Ces probabilités p_{ji} appelées *probabilités de transition* sont réunies dans la *matrice de transition* suivante :

$$P = \begin{pmatrix} p_{11} & p_{21} \\ p_{12} & p_{22} \end{pmatrix} = \begin{pmatrix} p_{11} & 1 - p_{22} \\ 1 - p_{11} & p_{22} \end{pmatrix}$$

avec $p_{ji} = P(S_t = i | S_{t-1} = j)$ la proba d'être dans l'état i sachant que l'on était dans l'état j à la période précédente.

- **Définition** : On appelle processus autorégressif à changement de régime markovien d'ordre p_1 et p_2 , noté MS-AR(p_1, p_2), un processus Y_t :

$$Y_t = \begin{cases} c_1 + \phi_{1,1} Y_{t-1} + \dots + \phi_{p_1,1} Y_{t-p_1} + \sigma_1 \varepsilon_t & \text{si } S_t = 1 \\ c_2 + \phi_{1,2} Y_{t-1} + \dots + \phi_{p_2,2} Y_{t-p_2} + \sigma_2 \varepsilon_t & \text{si } S_t = 2 \end{cases}$$

avec $t = 1, \dots, T$, ou de façon équivalente :

$Y_t = c_{S_t} + \phi_{1,S_t} Y_{t-1} + \dots + \phi_{p_{S_t},S_t} Y_{t-p_{S_t}} + \sigma_{S_t} \varepsilon_t$ $t = 1, \dots, T$
 avec ε_t i.i.d. $(0, 1)$ et S_t une variable d'état inobservable prenant 2 valeurs 1 ou 2 suivant une chaîne de Markov d'ordre un de matrice de transition entre états donnée par :

$$P = \begin{pmatrix} p_{11} & p_{21} \\ p_{12} & p_{22} \end{pmatrix} = \begin{pmatrix} p_{11} & 1 - p_{22} \\ 1 - p_{11} & p_{22} \end{pmatrix}$$

- **Remarque** : Pour simplifier, on considèrera souvent que $p_1 = p_2 = p$.
- **Commentaire** : Contrairement aux modèles à seuil, l'état qui se réalise à la date t n'est pas observé. On sait seulement qu'il existe deux états (ou plus) et qu'à tout instant, le système a une certaine probabilité de rester dans l'état i dans lequel il se trouve p_{ii} et la probabilité complémentaire d'en sortir $(1 - p_{ii})$.
- **Remarque** : dans le cas où $p = 1$ et si ε_t i.i.d $N(0, 1)$, $Y_t | S_t = i, I_{t-1} \sim \mathcal{N}(c_i + \phi_i Y_{t-1}, \sigma_i^2)$ $i = \{1, 2\}$. On a donc :

$$f(Y_t | S_t = i, I_{t-1}; \theta) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left\{ -\frac{(Y_t - c_i - \phi_i Y_{t-1})^2}{2\sigma_i^2} \right\}$$

Propriétés

- **Propriété** : Les probabilités *inconditionnelles* ou *ergodiques* des deux états sont données par :

$$P(S_t = 1) = \frac{1 - p_{22}}{2 - p_{11} - p_{22}} \text{ et } P(S_t = 2) = \frac{1 - p_{11}}{2 - p_{11} - p_{22}} \quad \forall t$$

Voir Hamilton 1994 pour une démonstration de ce résultat.

- **Propriété** : Les durées moyennes des régimes 1 et 2 sont données respectivement par $(1 - p_{11})^{-1}$ et $(1 - p_{22})^{-1}$.
- **Exemple** : Si $p_{11} = 0,9$, la durée moyenne du régime 1 est de 10 périodes.
- **Explication** : Soit D_i une variable aléatoire définie sur \mathbb{N}^* et désignant la durée du régime i ou de façon équivalente le nombre de périodes nécessaires pour sortir du régime i . Cette variable suit une loi géométrique de paramètre $p_{ij} = 1 - p_{ij}$. On utilise alors les résultats théoriques sur les moments de cette loi.

Démonstration

L'espérance d'une variable aléatoire X distribuée suivant une loi géométrique de paramètre p est donnée par :

$$E(X) = \sum_{k=1}^{\infty} kP(X=k) = p \sum_{k=1}^{\infty} k(1-p)^{k-1}$$

Or $\sum_{k=0}^N x^k = \frac{1-x^{N+1}}{1-x} \xrightarrow{N \rightarrow \infty} \frac{1}{1-x}$ si $|x| < 1$

On en déduit que si $|x| < 1$, $\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$

On dérive cette somme par rapport à x :

$$\frac{\partial}{\partial x} \left(\sum_{k=0}^{\infty} x^k \right) = \frac{\partial}{\partial x} \left(\frac{1}{1-x} \right) \Leftrightarrow \sum_{k=0}^{\infty} kx^{k-1} = \sum_{k=1}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}$$

On obtient donc $E(X) = p \frac{1}{(1-(1-p))^2} = \frac{p}{p^2} = \frac{1}{p}$ ■

Probabilités filtrées et lissées

Les états en $t = 1, \dots, T$ ne sont pas observés, mais il est possible d'estimer leur probabilité de réalisation sur la base de l'information observée présente et passée (*probabilités filtrées*) ou sachant l'ensemble de l'information dont on dispose (*probabilités lissées*).

- **Définition** : On appelle *probabilité filtrée* du régime i à l'instant t la probabilité de se situer dans le régime i en t conditionnellement à l'information I_t dont on dispose à l'instant t :

$$P(S_t = i | Y_t, Y_{t-1}, \dots, Y_1; \theta) = P(S_t | I_t; \theta)$$

où $\theta = (p_{11}, p_{22}, \phi'_1, \phi'_2, \sigma_1, \sigma_2)$ et $\phi'_j = (c_j, \phi_{1,j}, \dots, \phi_{p,j})$,
 $j = 1, 2$.

Probabilités filtrées et lissées

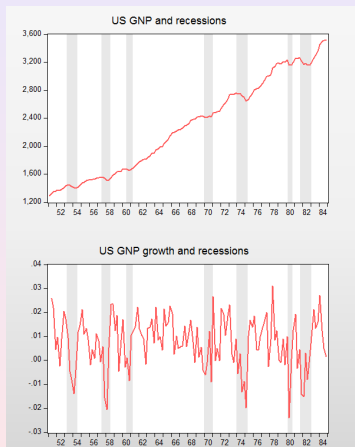
- **Définition** : On appelle *probabilité lissée* du régime i à l'instant t la probabilité de se situer dans le régime i en t conditionnellement à l'ensemble de l'information I_T contenue dans l'échantillon :

$$P(S_t = i | Y_T, Y_{T-1}, \dots, Y_1; \theta) = P(S_t | I_T; \theta)$$

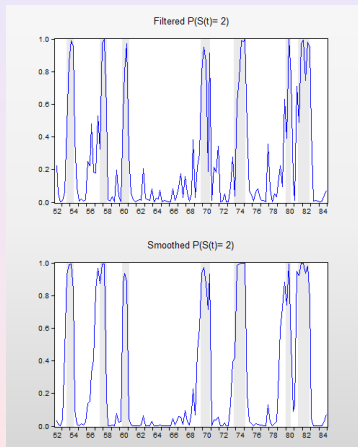
où $\theta = (p_{11}, p_{22}, \phi'_1, \phi'_2, \sigma_1, \sigma_2)$ et $\phi'_j = (c_j, \phi_{1,j}, \dots, \phi_{p_j,j})$, $j = 1, 2$.

Elles fournissent grâce au surcroît d'information résultant de l'utilisation de l'ensemble de l'échantillon une meilleure information sur la nature de l'état aux différents instants du temps considérés.

Illustration : datation du cycle



(a) Input



(b) Output

Identification ou choix des variables explicatives

- **MS-AR** : Pour simplifier, on utilisera souvent l'ordre p déterminé pour le processus autorégressif linéaire suivant la méthodologie de Box et Jenkins (on utilise le corrélogramme partiel et/ou les critères d'information).
- **Modèle de régression quelconque** : Les tests de significativité habituels des coefficients peuvent être appliqués pour éliminer les variables qui ne sont pas pertinentes.

Pour simplifier, considérons le MS-AR(1,1) suivant :

$$Y_t = c_{S_t} + \phi_{S_t} Y_{t-1} + \sigma_{S_t} \varepsilon_t \quad t = 2, \dots, T$$

avec $S_t = (1, 2)$. Le vecteur de paramètres à estimer est $\theta = (p_{11}, p_{22}, c_1, c_2, \phi_1, \phi_2, \sigma_1, \sigma_2)$

La présence d'une variable d'état inobservable dans l'équation empêche l'application des méthodes d'estimation usuelles.

Deux méthodes d'estimation peuvent être appliquées :

- 1 maximiser la vraisemblance obtenue dans un algorithme de filtrage ;
- 2 utiliser l'algorithme espérance-maximisation qui consiste à calculer et maximiser l'espérance de la vraisemblance du modèle.

Nous exposons ici la première méthode.

Principe général

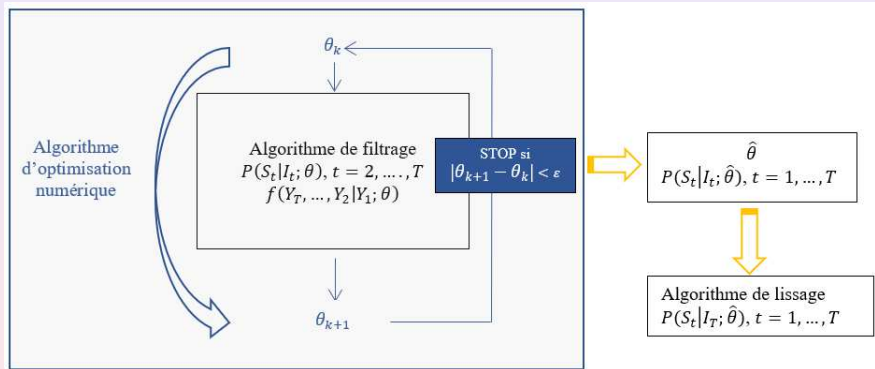


Figure – Principe général de la méthode d'estimation

Soit θ_1^* un vecteur de paramètres initial

- 1 **Algorithme de filtrage** et calcul de la vraisemblance du modèle (conditionnelle aux $p = 1$ premières observations) :

$$P(S_t = i | Y_t, Y_{t-1}, \dots; \theta_1^*) = P(S_t = i | I_t; \theta_1^*) \text{ pour } t = p + 1, \dots, T$$

$$f(Y_T, Y_{T-1}, \dots, Y_2 | Y_1; \theta_1^*)$$
- 2 **Algorithme d'optimisation numérique** (comme Newton Raphson) pour obtenir une nouvelle valeur du vecteur de paramètres $\theta_2^* : \theta_2^* = \theta_1^* - \left[\frac{d^2 f}{d\theta^2}(\theta_1^*) \right]^{-1} \frac{df}{d\theta}(\theta_1^*)$ avec f la vraisemblance du modèle.
- 3 Réintroduction de la nouvelle valeur θ_2^* dans l'**algorithme de filtrage** et ainsi de suite jusqu'à ce que $|\theta_{k+1}^* - \theta_k^*| < \varepsilon$. On notera θ^* le vecteur de paramètres finalement obtenu
- 4 **Algorithme de lissage** en θ^* pour affiner l'estimation des états aux différents instants du temps :

$$P(S_t = i | Y_T, Y_{T-1}, \dots; \theta^*) = P(S_t = i | I_T; \theta^*) \text{ pour } t = 1, \dots, T$$

Algorithme de filtrage

La construction des probabilités filtrées et de la vraisemblance des observations se fait en cinq étapes itérées de $t = p + 1$ à $t = T$, étant données les conditions initiales y_1, \dots, y_p et une valeur initiale du vecteur de paramètres θ .

- ❶ Etape 1 : Calcul de la probabilité jointe $P(S_t, S_{t-1} | I_{t-1}; \theta)$:

$$P(S_t = i, S_{t-1} = j | I_{t-1}; \theta) =$$

$$P(S_t = i | S_{t-1} = j) \times P(S_{t-1} = j | I_{t-1}; \theta) \quad i, j = \{1, 2\},$$

On pourra utiliser comme valeurs initiales $P(S_1 = j | I_1; \theta)$, $j = \{1, 2\}$ les probabilités ergodiques

- ❷ Etape 2 : Calcul de la probabilité $P(S_t | I_{t-1}; \theta)$:

$$P(S_t = i | I_{t-1}; \theta) = \sum_{j=1}^2 P(S_t = i, S_{t-1} = j | I_{t-1}; \theta), \quad i = \{1, 2\}$$

Algorithme de filtrage

- ③ Etape 3 : Calcul de la densité jointe $f(Y_t, S_t | I_{t-1}; \theta)$:
- $$f(Y_t, S_t = i | I_{t-1}; \theta) = f(Y_t | S_t = i, I_{t-1}; \theta) \times P(S_t = i | I_{t-1}; \theta)$$
- $i = \{1, 2\}$, où sous l'hypothèse de normalité des résidus :

$$f(Y_t | S_t = i, I_{t-1}; \theta) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left\{ -\frac{(Y_t - c_i - \phi_i Y_{t-1})^2}{2\sigma_i^2} \right\}$$

- ④ Etape 4 : Calcul de la densité conditionnelle $f(Y_t | I_{t-1}; \theta)$
- $$f(Y_t | I_{t-1}; \theta) = \sum_{i=1}^2 f(Y_t, S_t = i | I_{t-1}; \theta), i = \{1, 2\}$$

- ⑤ Etape 5 : Calcul de la probabilité filtrée $P(S_t | I_t; \theta)$ qui sera utilisée à l'itération suivante du filtre

$$P(S_t = i | I_t; \theta) = \frac{f(Y_t, S_t = i | I_{t-1}; \theta)}{f(Y_t | I_{t-1}; \theta)}, i = \{1, 2\}$$

Algorithme de filtrage

La vraisemblance totale du processus conditionnelle aux $p = 1$ premières observations est calculable à l'issue des T itérations du filtre en multipliant les densités conditionnelles obtenues à l'étape 4 du filtre

$$\begin{aligned} & f(Y_T, Y_{T-1}, \dots, Y_2 | Y_1; \theta) \\ &= f(Y_T | I_{t-1}; \theta) \times \dots \times f(Y_4 | I_3; \theta) \times f(Y_3 | I_2; \theta) \times f(Y_2 | I_1; \theta) \end{aligned}$$

Algorithme d'optimisation numérique

On cherche le vecteur de paramètres θ qui maximise la vraisemblance $f(Y_T, Y_{T-1}, \dots, Y_2 | Y_1; \theta)$.

Le problème d'optimisation n'admettant pas de solution analytique, on utilise un algorithme d'optimisation numérique de type Newton Raphson (voir chapitre 3). Il s'agira d'améliorer l'approximation suivante au cours d'itérations successives :

$$\theta_{k+1}^* = \theta_k^* - \left[\frac{d^2 f}{d\theta^2}(\theta_k^*) \right]^{-1} \frac{df}{d\theta}(\theta_k^*)$$

avec f la vraisemblance conditionnelle du modèle.

On relancera la procédure d'estimation pour plusieurs vecteurs de paramètres initiaux θ_0 car rien n'assure que le vecteur $\hat{\theta}$ obtenu ne soit pas un optimum local.

Algorithme de lissage

Une fois les paramètres du modèle estimés, on peut calculer, en l'optimum obtenu $\hat{\theta}$, les **probabilités lissées** $P(S_t | I_T; \theta)$ pour $t = 1, \dots, T$ avec $I_T = (Y_T, Y_{T-1}, \dots, Y_1)$ grâce à l'**algorithme de lissage**.

Grâce au surcroît d'information résultant de l'utilisation de l'ensemble de l'échantillon, les probabilités lissées fournissent une meilleure information sur la nature de l'état aux différents instants du temps considérés.

Algorithme de lissage

Hypothèse simplificatrice : on supposera que S_{t+1} résume toute l'information disponible concernant le futur :

$$P(S_t = i \mid S_{t+1} = j, Y_T, \dots, Y_1; \theta) = P(S_t = i \mid S_{t+1} = j, Y_t, \dots, Y_1; \theta)$$

Sous cette hypothèse, les probabilités lissées sont obtenues en itérant en arrière *i.e.* pour $t = T - 1$ à 1 le calcul suivant :

$$P(S_t = i \mid Y_T, \dots, Y_1; \theta) = \sum_{j=1}^2 \frac{P(S_{t+1}=j \mid S_t=i) \times P(S_t=i \mid Y_t, \dots, Y_1; \theta)}{P(S_{t+1}=j \mid Y_t, \dots, Y_1; \theta)} \times P(S_{t+1} = j \mid Y_T, \dots, Y_1; \theta)$$

Remarque : l'algorithme est initialisé par le produit de la dernière étape du filtre $P(S_t = i \mid Y_T, \dots, Y_1; \theta)$ pour $t = T$.

Algorithme de lissage - démonstration

$$P(S_t = i \mid I_T; \theta) = \sum_{j=1}^2 \frac{P(S_{t+1} = j, S_t = i \mid I_t; \theta)}{P(S_{t+1} = j \mid I_t; \theta)} \times P(S_{t+1} = j \mid I_T; \theta)$$

$$\iff P(S_t = i \mid I_T; \theta) = \sum_{j=1}^2 P(S_t = i \mid S_{t+1} = j, I_t; \theta) \times P(S_{t+1} = j \mid I_T; \theta)$$

En utilisant l'hypothèse posée plus haut, on obtient encore :

$$\iff P(S_t = i \mid I_T; \theta) = \sum_{j=1}^2 P(S_t = i \mid S_{t+1} = j, I_T; \theta) \times P(S_{t+1} = j \mid I_T; \theta)$$

$$\iff P(S_t = i \mid I_T; \theta) = \sum_{j=1}^2 P(S_t = i, S_{t+1} = j \mid I_T; \theta) \blacksquare$$

- ❶ **Test de linéarité** avec les valeurs critiques obtenues par simulation suivant la démarche proposée par Hansen (1996), voir chapitre 3.
- ❷ **Critères d'information** : les critères classiques AIC et Schwarz selon Psaradakis et Spagnolo (Journal of Time Series Analysis, 2009) ou des critères d'information adaptés pour les modèles MS par Smith, Naik et Tsai (J. of Econometrics, 2006).

Note : Le test de Carrasco, Hu et Ploberger (Econometrica, 2014) permet de repérer une instabilité des paramètres mais ne permet pas de tester le nb de régimes ($H_0 = ML$).

Pb : Les résidus dépendent de la variable d'état inobservable et sont donc eux-mêmes inobservés. Une solution simple repose sur les résidus généralisés introduits par Gouriéroux, Montfort, Trognon et Renault (1987) dans le cadre des modèles à variables latentes (probit, tobit. .).

Les **résidus généralisés** sont obtenus comme suit dans le cas de notre modèle MS-AR(p_1, p_2) :

$$e_t = \sum_{i=1}^2 \left(\frac{y_t - c_i - \phi_{1,i}y_{t-1} - \dots - \phi_{p_i,i}y_{t-p_i}}{\sigma_i} \right) P(S_t = i \mid I_{t-1}; \theta)$$

pour $t = 1, \dots, T$ avec $P(S_t = i \mid I_{t-1}; \theta)$ la probabilité prévue de l'état calculée à l'étape 2 du filtre.

On appliquera ensuite le **test de Jarque Bera** (test de normalité) et le **test de Ljung Box** aux résidus généralisés e_t (test d'AR) ou aux résidus au carré e_t^2 (test d'effets ARCH).

- 1 Remarque : approche alternative plus complexe proposée par Hamilton (1996)
- 2 Remarque : dans le contexte des modèles à changements de régime markovien, Smith (Journal of Time Series Analysis 2008) utilise la transformation de Rosenblatt (1952) des résidus :

$$\tilde{e}_t = \Phi^{-1} \left(\sum_{i=1}^2 \Phi \left(\frac{y_t - c_i - \phi_i y_{t-1}}{\sigma_i} \right) P(S_t = i \mid I_{t-1}; \theta) \right)$$

avec Φ la fonction de répartition de la loi $N(0, 1)$. Smith (2008) montre que les tests ont de bonnes performances en échantillon fini pour la détection de l'autocorrélation sérielle sur la base d'expériences de Monte Carlo.

Extensions possibles

- Probabilités de transition variables en fonction de variables observées (Filardo, 1994, Diebold et al., 1994, Kim, Piger et Startz, 2008, Chang, Choi et Park, 2017)
- Probabilités de transition variables en fonction du score de la (log) densité conditionnelle (Bazzi, Blasques, Koopman et Lucas, 2017)
- Plusieurs chaînes de Markov (McConnell et Perez-Quiros, AER 2000 ; Chauvet et Potter, 2013)
- Modèle MS-MIDAS (Guérin et Marcellino, JBES 2013)
- MS-VAR (Krolzig, 1997)
- Modèle MS à facteurs (Kim et Nelson, 98, Chauvet 1998)
- Modèles MS sur des variables qualitatives \rightsquigarrow IR de l'Insee (Gregoir et Lengart, 2000 ; Bardaji et Tallet, 2009)

Spécification

Définition : On appelle processus MS-AR(p_1, p_2) à probabilités de transition variables, un processus Y_t :

$$Y_t = c_{S_t} + \phi_{1,S_t} Y_{t-1} + \dots + \phi_{p_{S_t}, S_t} Y_{t-p_{S_t}} + \sigma_{S_t} \varepsilon_t$$

où ε_t i.i.d (0, 1) et S_t est une variable d'état suivant une chaîne de Markov d'ordre un dont les probabilités de transition sont fonction d'une variable *exogène* Z_t :

$$P(S_t = i \mid S_{t-1} = j, Z_t) = P(S_t = i \mid S_{t-1} = j, S_{t-2} = k, \dots, Z_t, I_{t-1}) \\ = p_{ji}(Z_t)$$

La matrice de transition est de la forme :

$$\begin{pmatrix} p_{11}(Z_t) & 1 - p_{22}(Z_t) \\ 1 - p_{11}(Z_t) & p_{22}(Z_t) \end{pmatrix}$$

Spécifications possibles des probabilités de transition

1 Fonction logistique :

$$p_{11}(Z_t) = \frac{\exp(\alpha_0 + \alpha_1 Z_t)}{1 + \exp(\alpha_0 + \alpha_1 Z_t)} \text{ et } p_{22}(Z_t) = \frac{\exp(\beta_0 + \beta_1 Z_t)}{1 + \exp(\beta_0 + \beta_1 Z_t)}$$

2 Fonction de répartition de la loi $N(0,1)$:

$$p_{11}(Z_t) = F_0(\alpha_0 + \alpha_1 Z_t) \text{ et } p_{22}(Z_t) = F_0(\beta_0 + \beta_1 Z_t)$$

$$\text{avec } F_0(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt$$

Remarque : Ces formes fonctionnelles assurent que les probabilités soient comprises entre 0 et 1.

Test du modèle PTF versus PTV

Test du modèle à probabilités de transition fixes (PTF) contre le modèle à probabilités de transition variables (PTV) : test du rapport de vraisemblance standard

- Hypothèses : $H_0 : \alpha_1 = \beta_1 = 0$ (PTF) contre $H_1 : \overline{H_0}$ (PTV)
- Statistique de test : $LR = 2[L(\hat{\theta}_{PTV}) - L(\hat{\theta}_{PTF})] \underset{H_0}{\sim} \chi^2_2$ avec
 $L(\hat{\theta}_{PTF})$: la log-vraisemblance du modèle sous H_0
(modèle à PTF) et $L(\hat{\theta}_{PTV})$: la log-vraisemblance sous H_1 (modèle à PTV).
- Rejet de H_0 si $LR > \chi^2_{2,1-\alpha}$

Estimation

- Le vecteur de paramètres à estimer contient deux paramètres supplémentaires par rapport au modèle PTF

$$\theta = (\alpha_0, \alpha_1, \beta_0, \beta_1, c_1, c_2, \phi_1, \phi_2, \sigma_1, \sigma_2)$$

- On estime de nouveau θ par maximum de vraisemblance avec la vraisemblance obtenue dans l'algorithme de filtrage.
- Les algorithmes de filtrage et de lissage reposent sur des calculs similaires avec les probabilités de transition variables (dans l'étape 1 du filtre et dans la formule de l'algorithme de lissage, les probabilités de transition sont écrites en fonction de la variable Z_t)

Applications

- Cycle des affaires : modèle du taux de croissance de l'activité avec des probabilités de transition fonction d'indicateurs avancés du cycle économique (l'indice composite d'indicateurs avancés de Stock et Watson, le spread de taux d'intérêt, l'indice S&P, etc) dans Filardo (1994), Bessec (2019).
- Prix de l'immobilier : modèle avec un régime de bulle avec probabilité d'éclatement de la bulle spéculative fonction de la taille de la bulle (Hall, Psaradakis et Sola, 1997).
- Marché des changes : modèle chartiste et fondamentaliste avec des probabilités de transition fonction de l'occurrence des interventions des autorités monétaires sur le marché des changes (Beine, de Grauwe et Grimaldi, 2009)
- Prix du pétrole et spread souverain de pays exportateurs de pétrole (Chuffart et Hooper, 2019)

Spécification

Reprenons notre processus MS-AR dont la dynamique dépend maintenant de 2 variables d'état :

$$Y_t = c_{S_t^A} + \phi_{1, S_t^A} Y_{t-1} + \dots + \phi_{p_{S_t^A}, S_t^A} Y_{t-p_{S_t^A}} + \sigma_{S_t^B} \varepsilon_t \quad t = 1, \dots, T$$

avec ε_t i.i.d(0,1), $S_t^A = \{1, 2\}$ et $S_t^B = \{1, 2\}$ deux chaînes de Markov d'ordre un de matrices de transition P^A et P^B .

$$P^A = \begin{pmatrix} p_{11}^A & 1 - p_{22}^A \\ 1 - p_{11}^A & p_{22}^A \end{pmatrix} \text{ et } P^B = \begin{pmatrix} p_{11}^B & 1 - p_{22}^B \\ 1 - p_{11}^B & p_{22}^B \end{pmatrix}$$

Le mécanisme qui régit le changement de régime de la volatilité diffère de celui de la moyenne et de la dynamique autorégressive.

Spécification

Ecriture équivalente : modèle standard à 4 états avec une seule variable d'état suivant une chaîne de Markov d'ordre un et définie comme suit :

$$S_t = \begin{cases} 1 & \text{si } S_t^A = 1, S_t^B = 1 \\ 2 & \text{si } S_t^A = 2, S_t^B = 1 \\ 3 & \text{si } S_t^A = 1, S_t^B = 2 \\ 4 & \text{si } S_t^A = 2, S_t^B = 2 \end{cases}$$

La matrice de transition associée est de dimension 4...

La matrice de transition associée est de dimension 4 :

$$P = \begin{pmatrix} p_{11} & p_{21} & p_{31} & p_{41} \\ p_{12} & p_{22} & p_{32} & p_{42} \\ p_{13} & p_{23} & p_{33} & p_{43} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{pmatrix}$$

$$= \begin{pmatrix} p_{11} & p_{21} & p_{31} & p_{41} \\ p_{12} & p_{22} & p_{32} & p_{42} \\ p_{13} & p_{23} & p_{33} & p_{43} \\ 1 - \sum_{i=1}^3 p_{1i} & 1 - \sum_{i=1}^3 p_{2i} & 1 - \sum_{i=1}^3 p_{3i} & 1 - \sum_{i=1}^3 p_{4i} \end{pmatrix}$$

Cas particuliers

Deux cas particuliers sont généralement étudiés

- ❶ S_t^A et S_t^B indépendantes auquel cas $P = P^B \otimes P^A$

$$P = \begin{pmatrix} p_{11} & p_{21} & p_{31} & p_{41} \\ p_{12} & p_{22} & p_{32} & p_{42} \\ p_{13} & p_{23} & p_{33} & p_{43} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{pmatrix} = \begin{pmatrix} p_{11}^A p_{11}^B & (1 - p_{11}^A) p_{11}^B & p_{11}^A (1 - p_{22}^B) & (1 - p_{11}^A) (1 - p_{22}^B) \\ (1 - p_{11}^A) p_{11}^B & p_{22}^A p_{11}^B & (1 - p_{11}^A) (1 - p_{22}^B) & p_{22}^A (1 - p_{22}^B) \\ p_{11}^A (1 - p_{11}^B) & (1 - p_{22}^A) (1 - p_{11}^B) & p_{11}^A p_{22}^B & (1 - p_{22}^A) p_{22}^B \\ (1 - p_{11}^A) (1 - p_{11}^B) & p_{22}^A (1 - p_{11}^B) & (1 - p_{11}^A) p_{22}^B & p_{22}^A p_{22}^B \end{pmatrix}$$

Dans ce cas, il n'y a plus que 4 probabilités de transition à estimer (contre 16 dans le cas général).

Cas particuliers

- ② S_t^A et S_t^B parfaitement corrélées auquel cas on retrouve le modèle classique avec une seule chaîne de Markov (à deux états).

$$P = \begin{pmatrix} p_{11} & - & - & 1 - p_{44} \\ - & - & - & - \\ - & - & - & - \\ 1 - p_{11} & - & - & p_{44} \end{pmatrix}$$

Pour tester ces deux cas particulier contre la spécification générale, test du rapport de vraisemblance (non standard dans le second cas).

Remarque : à l'issue de l'estimation du modèle, on obtiendra les probabilités lissées de 4 états $P(S_t = i | I_T; \theta)$, $i = 1, \dots, 4$. On pourra en déduire les probabilités lissées des deux états de chaque chaîne :

$$P(S_t^A = 1 | I_T; \theta) = P(S_t = 1 | I_T; \theta) + P(S_t = 3 | I_T; \theta)$$

$$P(S_t^A = 2 | I_T; \theta) = P(S_t = 2 | I_T; \theta) + P(S_t = 4 | I_T; \theta)$$

$$P(S_t^B = 1 | I_T; \theta) = P(S_t = 1 | I_T; \theta) + P(S_t = 2 | I_T; \theta)$$

$$P(S_t^B = 2 | I_T; \theta) = P(S_t = 3 | I_T; \theta) + P(S_t = 4 | I_T; \theta)$$

Applications

- Cycle des affaires : McConnell et Perez-Quiros (2000), Chauvet et Potter (2013) pour filtrer l'influence de la volatilité dans l'identification des régimes d'expansion et de récession.
- Marché financiers : Bessec (2005) pour filtrer l'influence de la volatilité dans l'identification des régimes chartiste et fondamentaliste, Kole and van Dijk (JAE, 2017) pour filtrer l'influence de la volatilité dans l'identification des phases haussières et baissières du marché (*bear vs bull markets*).

Définition : Soit un processus MS-AR(p_1, p_2) :

$$Y_t = c_{S_t} + \phi_{1,S_t} Y_{t-1} + \cdots + \phi_{p_{S_t}, S_t} Y_{t-p_{S_t}} + \sigma_{S_t} \varepsilon_t$$

et supposons maintenant que S_t suit une chaîne de Markov d'ordre deux et non d'une chaîne de Markov d'ordre un.

$$\begin{aligned} & P(S_t = i \mid S_{t-1} = j, S_{t-2} = k) \\ &= P(S_t = i \mid S_{t-1} = j, S_{t-2} = k, S_{t-3} = l, \dots, l_{t-1}) = p_{kji} \quad \forall t \end{aligned}$$

avec $i, j, k = \{1, 2\}$ et l_{t-1} l'ensemble des variables exogènes connues à la date $t - 1$.

Ecriture équivalente avec une chaîne de Markov d'ordre un

Définissons alors la nouvelle variable d'état \tilde{S}_t :

$$\tilde{S}_t = \begin{cases} 1 & \text{si } S_t = 1, S_{t-1} = 1 \\ 2 & \text{si } S_t = 1, S_{t-1} = 2 \\ 3 & \text{si } S_t = 2, S_{t-1} = 1 \\ 4 & \text{si } S_t = 2, S_{t-1} = 2 \end{cases}$$

avec une matrice de transition dont certaines probabilités sont nulles :

$$P = \begin{bmatrix} \tilde{p}_{11} & \tilde{p}_{21} & \tilde{p}_{31} & \tilde{p}_{41} \\ \tilde{p}_{12} & \tilde{p}_{22} & \tilde{p}_{32} & \tilde{p}_{42} \\ \tilde{p}_{13} & \tilde{p}_{23} & \tilde{p}_{33} & \tilde{p}_{43} \\ \tilde{p}_{14} & \tilde{p}_{24} & \tilde{p}_{34} & \tilde{p}_{44} \end{bmatrix} \equiv \begin{bmatrix} p_{111} & p_{211} & 0 & 0 \\ 0 & 0 & p_{121} & p_{221} \\ 1 - p_{111} & 1 - p_{211} & 0 & 0 \\ 0 & 0 & 1 - p_{121} & 1 - p_{221} \end{bmatrix}$$

On retrouve donc une spécification classique que l'on estimera avec les méthodes présentées précédemment. Il faudra initialiser avec des probabilités de l'état en $t = 1$ et $t = 2$ et plus seulement en $t = 1$ avec une chaîne de Markov d'ordre un.

- Cycle des affaires (Hamilton, Econometrica 1989 ; Kim et Nelson, Rev. of Eco Stat 1999, Chauvet et Piger, JBES 2008)
- IR conjoncturel de l'Insee (Grégoir et Lengart, JoF 2000, Ast, 2010, Bardaji et Tallet, 2009)
- Détection des bulles spéculatives (Hall Pasaradakis et Sola, Economic Modelling 1997, JAE 1999, Driffill et Sola JME 1998, Anaswah, Wilfling, JBF 2011)
- Taux de change (Engel et Hamilton, AER 1990 ; Beine, de Grauwe et Grimaldi, JBF, 2009)
- Taux d'intérêt (Gray, J. of Fin. Econ. 1996, Ang et Beckaert, JBES 2002, Davig et Doh, Rev. Eco Stat 2014)
- Rendements boursiers (Chiarella et al., JEBO 2012 ; Chiarella, He, Zwinkels, JEBO 2014)

ECONOMETRIE NON LINEAIRE

Chapitre 5. Apprentissage automatique ou *Machine learning*

M2 Economie - Université Paris-Dauphine

Sommaire et références

Plan du chapitre

- 1 Panorama des méthodes
- 2 Arbres de régression
- 3 Forêts aléatoires

Ouvrage de référence

[Hastie, T., Tibshirani, R., & Friedman, J. 2009.](#) In : The Elements of Statistical Learning : Data Mining, Inference, and Prediction. Springer-Verlag.

Il existe deux grandes catégories de méthodes :

- **Apprentissage supervisé** (*supervised learning*) : la cible est connue (exemple : taux de croissance du PIB, état du cycle) ; on entraîne des modèles sur un ensemble connu d'inputs (i.e un ensemble de variables explicatives) à prévoir un output (i.e. la cible à prévoir)
- **Apprentissage non supervisé** (*unsupervised learning*) : on cherche à identifier des caractéristiques des inputs pour partitionner les données ; il n'y a pas d'output donné pour guider l'apprentissage

On s'intéresse ici à l'apprentissage supervisé.

Apprentissage supervisé - régression

On prévoit ici une variable quantitative (par exemple, le taux de croissance du PIB)

Principales méthodes disponibles

- Régressions pénalisées (LASSO, LARS-EN, ridge)
- Support vector machine regressions
- Arbres de régression et forêts aléatoires (*decision tree and random forest*)
- Réseaux de neurones (*neural network*)
- Ensemble : weighted combination of different predictors

Apprentissage supervisé - classification

On prévoit ici une variable qualitative (par exemple, état de récession, défaillance d'une entreprise, spam, etc)

Principales méthodes disponibles

- La classification bayésienne naïve (*Naïve Bayes classifier*)
- Les *Support vector machines* (SVM)
- Arbres de classification et forêts aléatoires (*decision tree and random forest*)
- Réseaux de neurones (*neural network*)
- L'analyse discriminante (*Discriminant analysis*)
- Les méthodes des k plus proches voisins (*Nearest neighbors* kNN)

Apprentissage supervisé - Démarche générale

Les algorithmes recherchent la fonction des *inputs* qui prévoit le mieux l'*output* hors-échantillon.

Pour cela, les modèles sont entraînés sur une première sous-partie de l'échantillon (*train set*). Le choix des paramètres de réduction (*regularization*) qui contrôlent le niveau de complexité de la fonction (par exemple, la profondeur de l'arbre dans le cas des arbres de régression) et évitent un surajustement des modèles peut être affiné sur une seconde partie de l'échantillon (*evaluation set*). Les modèles retenus sont ensuite évalués sur la partie restante de l'échantillon (*test set*).

Une alternative à la régression linéaire

La régression linéaire $Y = X\beta + \varepsilon$ est un modèle global valide sur l'espace total des données. Quand les interactions entre les variables sont plus complexes, éventuellement non-linéaires, il devient compliqué de concevoir un modèle global unique.

Une alternative à la régression non-linéaire consiste à partitionner l'espace des variables explicatives en des régions plus petites où les interactions entre les variables sont plus simples. On partitionne ensuite les régions obtenues de nouveau jusqu'à ce que l'on obtienne des sous-espaces où l'on peut appliquer des modèles simples (en l'occurrence, une constante égale à la moyenne de y dans le sous-espace ou de façon équivalente, une régression linéaire de Y sur une constante pour les seules observations du sous espace = *piecewise regression model*).

Exemple des prix de l'immobilier

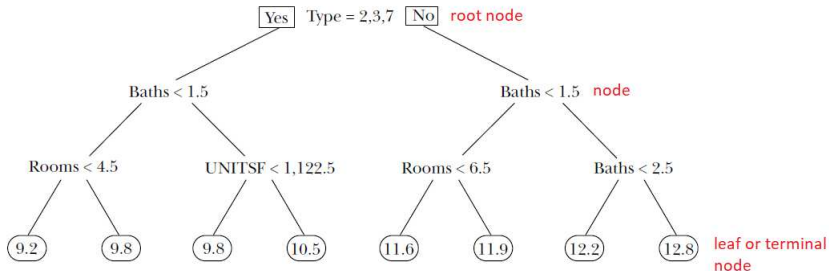
Dans le cas de la modélisation des prix de l'immobilier, il y a de nombreuses interactions entre les variables explicatives ; par exemple, l'effet du nombre de chambres dépend de la taille du logement, la présence d'un ascenseur de l'étage du logement, la taille du logement de la localisation du logement, etc.

Pour prendre en compte les nombreuses interactions entre les variables explicatives, il faudrait intégrer dans le modèle linéaire de nombreux produits de variables explicatives, ce qui n'est pas faisable car cela conduirait à un nombre de régresseurs supérieur au nombre d'observations.

Une alternative pour prendre en compte ces interactions consiste à utiliser un arbre de régression.

Exemple des prix de l'immobilier (suite)

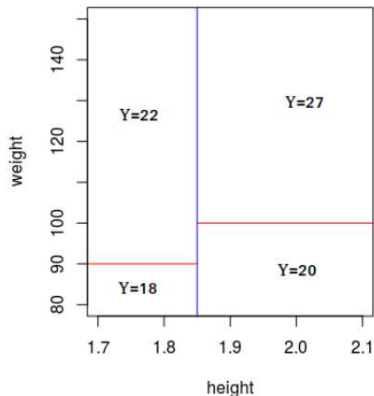
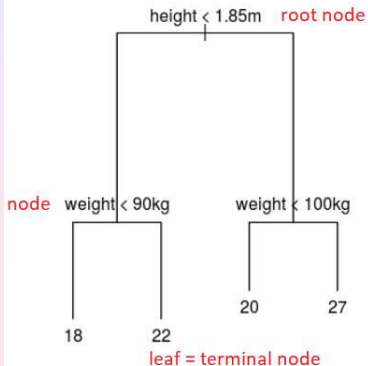
A Shallow Regression Tree Predicting House Values



Note: Based on a sample from the 2011 American Housing Survey metropolitan survey. House-value predictions are in log dollars.

Source : Mullainathan et Spiess (2017). Machine Learning : An Applied Econometric Approach. J. of Economic Perspectives.

Exemple : nb de points des basketteurs



Démarche générale

Soit y_i , la variable à prévoir (*output*) et x_{ij} , $j = 1, \dots, p$ les p variables explicatives potentielles (*inputs*). On dispose de $i = 1, \dots, N$ observations de ces variables.

On cherche la partition des valeurs de y en fonction des valeurs des x_j qui minimise S :

$$S = \sum_{c \in \text{leave}(T)} \sum_{i \in c} (y_i - m_c)^2$$

avec $m_c = \frac{1}{n_c} \sum_{i \in c} y_i$ = prévision pour la feuille (*leaf*) c

Remarque : $S = \sum_{c \in \text{leave}(T)} n_c V_c$ avec V_c la variance de y dans la feuille c

Démarche générale : partition

- 1 On considère d'abord l'ensemble des N observations de y et on calcule m_c et S .
 - 2 Si aucune partition binaire des y_i suivant les valeurs d'une variable explicative ne diminue S , on s'arrête. Sinon, on balaye toutes les partitions binaires possibles des observations suivant chaque variable explicative et on retient celle qui réduit le plus S .
 - 3 Dans chaque nouveau sous-ensemble, on répète l'étape 1.
- Arrêter les récursions quand la baisse de S devient plus petite qu'un seuil δ ou quand un minimum d'observations q à l'intérieur de chaque nouveau sous-ensemble a été atteint.

Démarche générale : pruning

On peut aussi retenir $q = 1$ and $\delta = 0$, i.e. faire grandir l'arbre le plus possible puis élaguer l'arbre (*pruning*).

- 1 On divise de façon aléatoire l'ensemble des N observations en un *training set* et un *validation set* (par exemple, 50% training et 50% validation).
- 2 On applique l'algorithme précédent sur le *training set* seulement avec $q = 1$ and $\delta = 0$.
- 3 On utilise ensuite le *validation set* pour élaguer l'arbre (*pruning*). Pour chaque paire de feuilles issue d'un noeud commun (*parent node*), on calcule si la somme des carrés devient plus petite si on fusionne les deux feuilles (le noeud parent devient alors une feuille).

On répète cette opération jusqu'à ce que l'élagage ne réduise plus la somme des carrés sur le *validation set*.

Limites des arbres de régression

- Inconvénient des arbres de régression uniques : instabilité des résultats (sensibilité à une légère variation des données ou à l'ordre des variables)
- Pour pallier ce problème, on pourra utiliser des forêts aléatoires. L'idée est d'effectuer un apprentissage sur de multiples arbres de régression entraînés sur des sous-ensembles de données légèrement différents.
- Inconvénient de la méthode : on perd l'aspect visuel des arbres de régression uniques

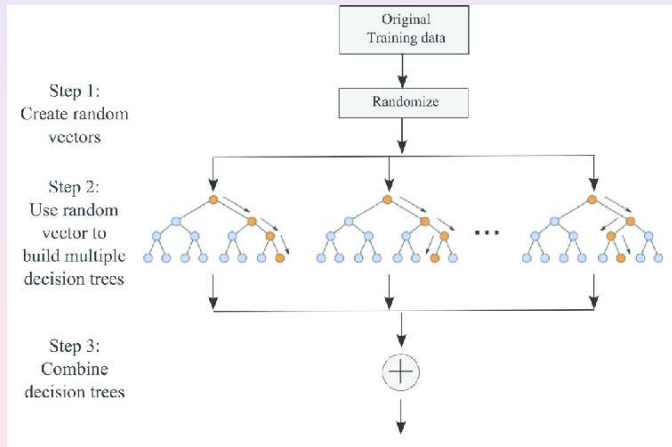
Démarche générale

La forêt aléatoire résout l'instabilité des arbres de régression en estimant des arbres de décision sur des échantillons de données légèrement différents.

Ces échantillons de données légèrement différents sont obtenus par *bootstrap* (tirage au hasard avec remise parmi les $i = 1, \dots, N$ observations) et en ne considérant qu'une partie des $j = 1, \dots, p$ variables également par tirage au hasard. On estime un arbre de régression sur chacun des échantillons ainsi constitués. La prévision finale est obtenue par moyenne simple des prévisions de chaque arbre.

Remarque : si on utilise la totalité des variables dans l'estimation des arbres, on ne parle plus de *random forest* mais simplement de *bagging* (pour *Bootstrap aggregating*).

Démarche générale



La méthode générale est la suivante :

- Créer B nouveaux ensembles d'apprentissage par un double processus d'échantillonnage :
 - ▷ sur les observations, en tirant avec remise un nombre N d'observations identique à celui des données d'origine (technique connue sous le nom de bootstrap),
 - ▷ sur les p prédicteurs, en n'en retenant qu'une proportion (1/3 par défaut sous matlab).
- Sur chaque échantillon, on entraîne un arbre de régression comme on l'a vu plus haut.
- On stocke les B prévisions de la variable d'intérêt pour chaque observation y_i .
- La prévision de la forêt aléatoire pour une observation y_i est alors obtenue simplement comme la moyenne des B prévisions de l'observation y_i .

- Rendements des actifs financiers (Kumar et al., 2014, Lohrmann et Luukka, 2019)
- Consommation d'électricité (Li et al., 2018, Ahmad et Chen, 2019, etc)
- Production d'énergie éolienne à un horizon d'un jour (Lahouar, Slama, 2017)
- Précipitations mensuelles (Chen et al., 2012)
- Prévision des émissions de CO₂ (Wei et al., 2018)
- Nowcasting de la croissance du PIB français (Veillon, note de conjoncture de l'Insee, 2019)

Autres références

Breiman L., J. Friedman, R. Olshen, and C. Stone.
Classification and Regression Trees. Boca Raton, FL : CRC
Press, 1984.

[Hastie, T., Tibshirani, R., & Friedman, J. 2009.](#) In : The
Elements of Statistical Learning : Data Mining, Inference, and
Prediction. Springer-Verlag.

Mullainathan S. et J. Spiess (2017). Machine Learning : An
Applied Econometric Approach. Journal of Economic
Perspectives.

Varian H.L. (2014). Big data : New tricks for econometrics.
Journal of Economic Perspectives.

TP1 : Simulation d'un processus AR, estimation et tests de diagnostic sous Matlab

Ouvrir un M-file dans lequel on stockera nos lignes de commandes

File \ New \ Script → Save as TP1.m

PARTIE 1 : LA PROGRAMMATION SOUS MATLAB

Remarques générales : pour exécuter un programme stocké dans un m-file, **touche F5** ; pour n'exécuter **qu'une sous-partie** du programme, **sélectionner les lignes de commandes et F9** ; Matlab distingue les **majuscules des minuscules** : $A \neq a$; des commentaires précédés du signe **%** peuvent être **insérés dans le code** (ils seront ignorés par Matlab) ; pour ne pas afficher le résultat d'une ligne, terminer cette ligne **par un point-virgule** ; on pourra couper une ligne de **commande trop longue** par **...**

- définition de vecteurs et matrices
 $A = [1 \ 3 ; 4 \ 5]$ ou $A = [1,3 ; 4,5]$
 $B = A ; B(2,3) = 0$
 $C = [A \ B] ; D = [A ; B]$
 $v = [1 ; 4 ; 5 ; 7]$
- **matrices particulières** : **ones(l,c)**, **zeros(l,c)**, **eye(l)** avec l : le nombre de lignes et c le nombre de colonnes de la matrice pour définir une matrice de 1, de 0 ou la matrice identité.
- $X = 0:0.5:4$ ou $x = \text{linspace}(0,4,9)$ génère le vecteur ligne $[0 \ 0.5 \ 1 \ 1.5 \ 2 \ 2.5 \ 3 \ 3.5 \ 4]$
- opérations sur les matrices :
 - o $\text{diag}(A)$: extrait la diagonale de la matrice A ; A' : transposée de la matrice A mais $\text{diag}(v)$ crée une matrice avec en diagonale le vecteur v et des 0 ailleurs
 - o $\text{det}(A)$, $\text{inv}(A)$, $\text{eig}(A)$, $\text{rank}(A)$ calculent respectivement le **déterminant de A**, **l'inverse de A**, le vecteur des valeurs propres de A (si la matrice A est carrée) et le rang de A .
 - o $\text{size}(A,1)$: **nombre de lignes de A**, $\text{size}(A,2)$: nombre de colonnes, $\text{size}(A)$: vecteur ligne contenant le nombre de lignes et le nombre de colonnes de A ;
 - o $\text{length}(v)$: nombre contenant le nombre de lignes d'un vecteur.
 - o $A+B$, $A-B$, $A*B$, $A.*B$, $A.^2$: **somme**, **différence**, **produit matriciel**, **produit élément par élément**, chaque élément élevé au carré (sans le point, produit matriciel de la matrice A par elle-même).
- Sélection des éléments d'une matrice : $A(i,j)$, $A(:,j)$, $A(i,:)$, $A(i,j:l)$ pour extraire respectivement l'élément (i,j) , la colonne j , la ligne i , les éléments à la ligne i et dans les colonnes j à l de A .
- Instructions conditionnelles

<pre>if condition expression elseif condition expression else expression end</pre>	<p><i>Exemple :</i></p> <pre>if A(1,1)>1 disp('la condition est vraie') else disp('la condition est fausse') end</pre>
--	---

Opérateurs pour définir une condition : égalité : $\text{if } a==b$; inégalité : $\text{if } a \sim b$; \geq : $\text{if } a \geq b$; ET : $\text{if } (a==b) \&\& (a==c)$; OU : $\text{if } a \sim b \&\& a \sim c$

- boucle

<pre>for i = valeur initiale:pas:valeur finale expression end</pre>	<p>% suite de Fibonacci</p> <pre>x = zeros(100,1); x(2)=1; for t = 3:100 x(t) = x(t-1) + x(t-2);</pre>
---	--

<pre>while condition expression end</pre>	<pre>end t=3; x = zeros(100,1); x(2)=1; while t <= 100 x(t) = x(t-1) + x(t-2) ; t=t+1 ; end</pre>
---	--

- fonction : l'utilisateur peut définir ses propres fonctions. Elles doivent être sauveées dans un m-File qui porte le nom de la fonction.

<pre>function y = mafonction(x) y = ...</pre>	<p>Enregistrer dans carre.m</p> <pre>function y = carre(x) y = x^2</pre>
---	--

Il suffit ensuite d'appeler *mafonction(x)* pour les valeurs de x désirées.

Remarque : toutes les variables utilisées à l'intérieur de la fonction sont locales (*i.e.* effacées de la mémoire lorsque l'on sort de la fonction). Inversement, les variables définies en dehors de la fonction sont globales, mais ne peuvent pas y être utilisées à moins de les mettre en argument de la fonction ou d'utiliser l'opérateur `global` (cf. TP3).

- graphique : *plot(x,y)*

Remarques de mise en forme

- o `plot(x,y,'a b c')` avec a : couleur (c cyan, m magenta, y jaune, r rouge, g vert, w blanc, k noir) , b : style de trait (- : continu, -- tirets, : pointillés, none : aucun), c : type de points (+, *, x, s : carré, d : diamant,...).
- o Pour rajouter un titre au graphique et aux axes :
`title('titre du graphe'); xlabel('légende abscisse'); ylabel('légende ordonnées');`
- o Pour rajouter une légende : `legend('courbe1', 'courbe2', 'courbe3',...)` ;
- o hold on pour que plusieurs commandes plot se superposent sur la même interface graphique (hold off pour désactiver cette commande).
- o Pour diviser l'interface graphique en plusieurs pavés, précéder chaque plot de subplot :
`subplot(nb lignes,nb colonnes, n° du graphe)`

- optimisation numérique de fonctions non-linéaires sous Matlab : fonction *fminunc*

Ces fonctions permettent d'obtenir le minimum d'une fonction lorsque le problème n'admet pas de solution analytique immédiate. Elles reposent sur des méthodes de résolution du type Newton-Raphson (cf. cours).

`[x,fval,exitflag,output,grad,hessian] = fminunc('nom_fonction',x0,options);`

Inputs :

- o `nom_fonction` : la fonction dont on recherche le minimum,
- o `x0` : le vecteur de paramètres initiaux,
- o `options` : options de l'algorithme d'optimisation (nombre d'itérations maximum, critères de convergence,...)

Outputs :

- o `x` : le vecteur solution qui minimise la fonction,
- o `fval` : la valeur de la fonction en x,
- o `exitflag` : code indiquant comment l'optimum a été atteint,
- o `output` : information sur l'optimisation (nb d'itérations, algorithme utilisé,...),
- o `grad` : les dérivées premières ou vecteur gradient de f calculées en la solution x,
- o `hessian` : la matrice hessienne ou matrice des dérivées secondes de f en x.

Options principales à définir avant l'appel de *fminunc* :

- o `Display` : gestion de l'affichage 'off' pas d'affichage ; 'iter' résultat affiché à chaque itération ; 'final' (default) affiche seulement le résultat final,

- **MaxIter** : nombre maximum d'itérations autorisées,
- **TolFun** : condition d'arrêt de l'algorithme sur la fonction objectif,
- **TolX** : condition d'arrêt de l'algorithme sur le vecteur de paramètres x .

Syntaxe : `options=optimset('ParameterName1',value1,'ParameterName2',value2,...)`

Exemple : `options = optimset('TolX',1e-12,'TolFun',1e-8,'Display','iter');`

Remarque : il existe d'autres fonctions d'optimisation comme **fminsearch** (on utilisera ici **fminunc** qui permet le calcul des dérivées premières et secondes par rapport à x).

Exercice

1. Créer les trois matrices suivantes :

$$A = \begin{pmatrix} -1 & 4 & 0 \\ 2 & 0 & 1 \\ 2 & 6 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 1 & 5 \\ 4 & 0 & 9 \\ -1 & 3 & 6 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

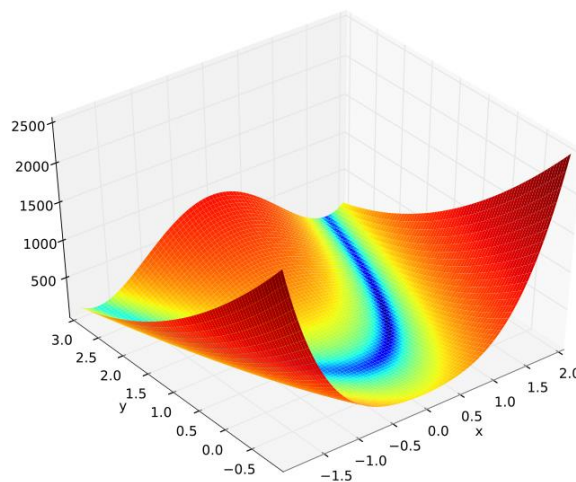
Que vaut $A*B$, $A.*B$, $A.^2$, $A>1$, $A(A>1)=0$? Stocker dans un vecteur ligne la 2^e ligne de A , puis dans un vecteur colonne la 1^{ère} colonne de B . Remplacer l'élément (2,2) de C par 0.

2. Créer la fonction f_1 telle que

$$f_1(x) = \begin{cases} 0 & \text{si } x \leq -1 \\ 2 - \exp(x^2/100) & \text{si } -1 < x \leq 3 \\ 1 & \text{si } x > 3 \end{cases}$$

Tester cette fonction en $x = 1.2$, puis $x = 5.7$.

3. Simuler 1000 observations d'une marche aléatoire en utilisant alternativement une boucle **for** et une boucle **while**. On utilisera le générateur de nombres gaussiens **randn**. Représenter graphiquement le résultat obtenu.
4. Représenter sur trois graphiques superposés $y_1 = \cos(x)$, $y_2 = \sin(x)$ et $y_3 = \exp(-x)$ avec x une suite de trente points répartis uniformément entre 0 et π . Indication : utiliser **subplot** et **linspace**.
5. Créer la fonction $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ (il s'agit de la fonction de Rosenbrock) et rechercher le vecteur $x^* = (x_1^* \ x_2^*)'$ qui minimise cette fonction. Le nombre d'itérations maximum sera fixé à 1000 et les critères d'arrêt de l'algorithme d'optimisation sur la fonction objectif et sur le vecteur solution à 10^{-10} .



PARTIE 2 : MANIPULATION D'UN PROCESSUS AR(1)

Exercice 1 : Simulation d'un processus AR(1)

On cherche à simuler un processus AR(1) défini comme suit :

$$X_t = c + \phi X_{t-1} + \sigma \varepsilon_t, \quad t = 2, \dots, T \text{ avec } \varepsilon_t \text{ i.i.d. } N(0,1)$$

où $c = 0.5$, $\phi = 0.8$, $\sigma = 0.9$ et $T = 500$.

- Simuler une **trajectoire de ce processus** en utilisant le **générateur de nombres gaussiens randn**. On stockera la trajectoire dans la variable **x**.
- Représenter la réalisation du processus **X_t** .

Exercice 2 : Identification de l'ordre du processus

En utilisant les fonctions **autocorr** et **parcorr** de Matlab, **représenter le corrélogramme et le corrélogramme partiel de la variable x**. Quel processus identifierait-on sur la base de ces deux graphiques ?

Facultatif. Retrouvez ce résultat en utilisant les critères d'information.

Exercice 3 : Estimation du processus AR(1)

Estimez les **paramètres c , ϕ et σ** du processus AR(1). Comparez les **estimations aux vraies valeurs des paramètres**.

Indications : On estimera par les **MCO** le vecteur $\beta = \begin{pmatrix} c \\ \phi \end{pmatrix}$ puis la variance résiduelle $\hat{\sigma}^2 = \frac{\sum_{t=2}^T e_t^2}{T-1}$ avec **e_t** les **résidus estimés du modèle**.

Exercice 4 : Tests de diagnostic

a) **Test de diagnostic sur les coefficients**. Vérifiez que la **constante** et que le coefficient ϕ sont significatifs.

Indications : 1) Calculer la matrice de **variance covariance** du vecteur de **paramètre β** donnée par $V(\hat{\beta}) = \hat{\sigma}^2(X'X)^{-1}$. Les variances estimées des paramètres figurent sur la diagonale de la matrice. 2) Calculer les **t-statistiques des coefficients c et ϕ** et comparer la **valeur absolue des t-statistiques à 1,96**. 3) En déduire si les coefficients sont **significativement différents de 0 au seuil de 5%**.

~~Facultatif. Calculer les probabilités critiques des deux tests.~~

b) **Test de diagnostic sur les résidus**. Stocker les résidus estimés du modèle estimé dans l'exercice 3 dans le vecteur **resid**.

En utilisant la commande matlab :

```
[h,pValue,stat,cValue] = lbqtest(resid,'lags',[5,10,20])
```

vérifiez l'absence d'autocorrélation à l'ordre **5, 10 et 20** dans les résidus. De même vérifiez l'absence d'autocorrélation dans les résidus au carré.

On pourra également **vérifier la normalité des résidus** en utilisant la commande :

```
[h,p,jbstat,critval] = jbtest(resid)
```

Facultatif. Retrouvez la valeur de la statistique de Ljung Box et de la statistique de Jarque Bera par le calcul.

Questions facultatives : application sur des données réelles

On considère à présent une série réelle constituée du taux de croissance trimestriel du PNB réel américain de 1990Q1 à 2016Q4 (source : <https://fred.stlouisfed.org/series/GDPMC1>) dans le fichier excel GDP_US.xlsx.

On demande d'identifier le processus autorégressif sur la base des corrélogrammes simple et partiel, d'estimer les paramètres du processus retenu et d'appliquer les tests de diagnostic usuels aux coefficients et aux résidus du modèle.

L'hypothèse de normalité des résidus est-elle acceptée ? Représenter graphiquement les résidus du modèle et proposer une solution simple pour améliorer la spécification.

TP2 : Simulation d'un processus SETAR, estimation et test sous Matlab

Objet : dans ce TP, on construit un programme permettant d'estimer le processus SETAR(1,1) défini pour $t = 1, \dots, T$ par $X_t = (c_1 + \phi_1 X_{t-1} + \sigma_1 \varepsilon_t) 1_{X_{t-1} \leq 0} + (c_2 + \phi_2 X_{t-1} + \sigma_2 \varepsilon_t) 1_{X_{t-1} > 0}$ avec ε_t i.i.d $N(0,1)$. On appliquera d'abord ce programme sur des données simulées suivant un processus SETAR(1,1) ; on vérifiera que la valeur des paramètres estimés est proche de la vraie valeur des paramètres du modèle et que les états estimés sont proches des états simulés. On utilisera ensuite ce programme pour mettre en évidence des effets de seuil dans la dynamique AR du PIB US en fonction de l'incertitude sur les marchés financiers.

Exercice 1 : Simulation d'un processus SETAR(1,1)

On cherche maintenant à simuler un processus SETAR(1,1) défini comme suit :

$$X_t = \begin{cases} c_1 + \phi_1 X_{t-1} + \sigma_1 \varepsilon_t & \text{si } X_{t-1} \leq 0 \\ c_2 + \phi_2 X_{t-1} + \sigma_2 \varepsilon_t & \text{si } X_{t-1} > 0 \end{cases} \quad t = 1, \dots, T$$

où $c_1 = -0.5$, $\phi_1 = 0.5$, $\sigma_1 = 1$ dans le régime 1 et $c_2 = 0.5$, $\phi_2 = 0.7$, $\sigma_2 = 1$ dans le régime 2 et avec ε_t i.i.d. $N(0,1)$.

- Simuler une trajectoire de ce processus de taille $T = 500$ en utilisant le générateur de nombres gaussiens `randn`.
- Représenter la réalisation du processus X_t et l'indicatrice de régime pour $t = 1, \dots, 500$.

Indication : pour empiler les deux graphiques, on utilisera la commande subplot de Matlab.

Exemple : `subplot(3,1,1),plot(x1)`; `subplot(3,1,2),plot(x2)`; `subplot(3,1,3),plot(x3)` ; divise l'interface graphique en trois et représente sur une colonne les graphiques de x_1 , x_2 et x_3 . La syntaxe est `subplot(nb lignes, nb colonnes, n° du graphe)`.

Exercice 2 : Identification de l'ordre du processus

En utilisant les fonctions `autocorr` et `parcorr` de Matlab, représenter le corrélogramme et le corrélogramme partiel de la variable x . Quel processus identifierait-on sur la base de ces deux graphiques ?

Facultatif. Retrouvez ce résultat en utilisant les critères d'information.

Exercice 3 : Estimation du processus SETAR

Estimer les paramètres du processus SETAR(1,1). Comparer les estimations aux vraies valeurs des paramètres. On pourra également comparer les états simulés aux états estimés.

Indications :

- Définir une variable `seuil_tri` qui contiendra les valeurs de seuil possibles triées par ordre croissant. *Indication : on appliquera la fonction `sort` à la variable de seuil.*
- Définir une variable `seuil_range` qui contiendra les valeurs de seuil possible. *Indication : on définira une nouvelle variable `seuil_range` en excluant les 15% plus petites valeurs et les 15% plus grandes valeurs de `seuil_tri`.*
- En utilisant une boucle sur toutes les valeurs de seuil possibles ainsi définies, estimer tous les modèles à seuil pour chacune des valeurs de seuil possibles et retenir le modèle qui minimise la variance résiduelle associée ; on stockera le seuil estimé dans la variable `seuil_opt`.

- Présenter les résultats d'estimation pour le seuil retenu et comparer le processus estimé au processus simulé.

Exercice 4 : Tests de diagnostic

a) *Tests de diagnostic sur les coefficients.* Vérifiez que les coefficients sont significatifs dans les deux régimes.

b) *Tests de diagnostic sur les résidus.* Stocker les résidus estimés du modèle estimé dans l'exercice 3 dans le vecteur **resid**.

En utilisant la commande matlab `lbqtest`, vérifiez l'absence d'autocorrélation dans les résidus. De même vérifiez l'absence d'autocorrélation dans les résidus au carré. On fera le test à l'ordre 5, 10 et 20. On pourra également vérifier la normalité des résidus en utilisant la commande `jbtest`.

Facultatif - Exercice 5 : Tests de linéarité

Calculer la **statistique de Wald** du test de linéarité, ainsi que sa **probabilité critique** et vérifier que **l'hypothèse nulle de linéarité est rejetée aux seuils usuels**.

Facultatif - application sur des données réelles

On souhaite étudier l'existence d'effet de seuil dans la dynamique AR du PIB américain en fonction de l'incertitude sur le marché financier américain. On utilisera de nouveau les données fournies dans le fichier GDP_US.xlsx (feuille GDP : taux de croissance trimestriel du PIB, feuille VIX : moyenne trimestrielle de l'indicateur VIX, mesure de la volatilité du marché financier US).

Pour mettre en évidence des effets de seuil, on estimera le TAR(p,p) suivant :

$$X_t = \begin{cases} c_1 + \phi_{1,1}X_{t-1} + \dots + \phi_{p,1}X_{t-p} + \varepsilon_t & \text{si } Z_t \leq \lambda \\ c_2 + \phi_{1,2}X_{t-1} + \dots + \phi_{p,2}X_{t-p} + \varepsilon_t & \text{si } Z_t > \lambda \end{cases} \quad t = p+1, \dots, T$$

avec le VIX comme variable de seuil Z_t .

On utilisera les critères d'information calculés dans le modèle à seuil pour le choix du nombre de retards p (on considèrera $p = 1, \dots, 4$). On estimera ensuite le modèle TAR pour l'ordre p retenu et l'on décrira les régimes obtenus. En utilisant un test de linéarité, on montrera que ce modèle est supérieur au processus AR linéaire considéré dans le TP1.

TP3 : Estimation d'un processus MS-AR sous Matlab

Objet : dans ce TP, on construit un programme permettant d'estimer le processus MS-AR(1,1) défini pour $t = 1, \dots, T$ par $Y_t = c_{S_t} + \phi_{1,S_t} Y_{t-1} + \sigma_{S_t} \varepsilon_t$ avec ε_t i.i.d $N(0,1)$ et S_t une variable d'état inobservable prenant 2 valeurs 1 ou 2 suivant une chaîne de Markov d'ordre un. On appliquera ce programme à un processus MS-AR(1,1) simulé ; on vérifiera que la valeur des paramètres estimés est proche de la vraie valeur des paramètres du modèle et que les états estimés sont proches des états simulés en comparant les réalisations effectives de la chaîne aux probabilités filtrées et lissées.

Structure générale du programme

1) PGM PAL : programme principal (*c'est ce programme qu'il faut exécuter*)

- il charge les données et définit les variables,
- il définit le vecteur de paramètres initiaux,
- il applique l'algorithme d'optimisation numérique à la fonction VRAIS
- il calcule les probabilités lissées pour le vecteur de paramètres optimaux avec LISSAGE

2) fonction VRAIS : calcule les probabilités filtrées et la fonction de vraisemblance,

Syntaxe : $LIKV = Vrais(x)$ avec x le vecteur de paramètres et $LIKV$ la valeur de la log-vraisemblance en x

3) fonction LISSAGE : calcule les probabilités filtrées puis les probabilités lissées.

Syntaxe : $[PL, PF] = Lissage(x)$ avec x le vecteur de paramètres et PL et PF les probabilités lissées et filtrées en x

I. Programme principal [Pgmpal.m]

a) Simulation des données

On simule d'abord un processus MS-AR(1) défini pour $t = 1, \dots, 200$ par :

$$Y_t = \begin{cases} -0,5 + 0,7Y_{t-1} + 0,5\varepsilon_t & \text{si } S_t = 1 \\ 0,5 + 0,5Y_{t-1} + \varepsilon_t & \text{si } S_t = 2 \end{cases}$$

avec ε_t i.i.d $N(0,1)$ et S_t une variable d'état inobservable suivant une chaîne de Markov d'ordre un à deux états et de matrice de transition entre états donnée par $P = \begin{pmatrix} 0,9 & 0,2 \\ 0,1 & 0,8 \end{pmatrix}$.

On utilise ici la fonction `Simul_MSAR`. On stocke dans le vecteur `endog` la variable Y_t simulée pour $t = 1, \dots, 200$, dans le vecteur `s` les états simulés pour $t = 1, \dots, 200$ et dans `x_dgp` la valeur des paramètres utilisés dans le processus générateur des données. Dans la suite, on supposera les états `s` et le vecteur de paramètres `x` inconnus ; on estimera `x` par maximum de vraisemblance et les états grâce aux probabilités filtrées et lissées.

b) Choix du vecteur de paramètres initiaux

Définir le vecteur `x0` des paramètres initiaux utilisé à la première itération de l'algorithme d'optimisation. Indications : utiliser le vecteur `x_dgp` auquel on rajoute un vecteur de perturbations aléatoires suivant une loi normale centrée et réduite. Pour ne pas avoir à contraindre l'espace de recherche des paramètres, ce qui n'est pas possible avec l'algorithme d'optimisation `fminunc`, on va

chercher $F_0^{-1}(p_{ii})$ plutôt que les probabilités de transition p_{ii} avec F_0 la fonction de répartition de la loi $N(0,1)$ par exemple. Sous Matlab, F_0 est donnée par `normcdf` et F_0^{-1} par `norminv`.

c) Estimation par MV des paramètres

- Définir les options de l'algorithme d'optimisation numérique : le nombre d'itérations maximum sera fixé à 1000 et les critères d'arrêt de l'algorithme d'optimisation sur la fonction objectif et sur le vecteur solution à 10^{-10} (cf. TP1).

- Faire appel à l'algorithme d'optimisation `fminunc` pour chercher le minimum de la fonction VRAIS. On stockera dans le vecteur `x` le vecteur de paramètres estimés, `fval` la vraisemblance au minimum, `code` le code diagnostic, `info` l'information fournie en sortie par `fminunc`, `g` le vecteur gradient et `H` la matrice hessienne (ou matrice d'information de Fisher).

- Calculer les écart-types estimés des paramètres estimés et en déduire les t-statistiques associées.

Indications : lorsque l'estimateur $\hat{\theta}$ de θ est efficace, la matrice de variance covariance de $\hat{\theta}$ est donnée par l'inverse de la matrice d'information de Fisher (borne de Frechet Darmois Cramer Rao) :

$$V(\hat{\theta}) = [I_n(\theta)]^{-1} \text{ avec } I_n(\theta) = E \left[\left(\frac{\partial \ln L}{\partial \theta} \right)^2 \right] = E \left[- \frac{\partial^2 \ln L}{\partial \theta^2} \right]$$

d) Affichage des résultats d'estimation

- Afficher le code diagnostic de retour indiquant si l'optimisation s'est correctement déroulée (voir la fin du programme pour l'interprétation du code diagnostic fourni en sortie par `fminunc`).

- Afficher la vraisemblance maximale estimée.

- Afficher en colonnes le vrai vecteur de paramètres `x_dgp`, le vecteur estimé `x`, les écart-types estimés et les t-statistiques.

- Afficher la matrice de transition estimée.

e) Représentation des probabilités filtrées lissées

Appeler la fonction `LISSAGE` en `x` et représenter graphiquement les vrais états `s`, les probabilités filtrées et les probabilités lissées.

Rappel : pour empiler les trois graphiques, on utilisera la commande `subplot` de Matlab (cf. TP1).

II. Algorithme de filtrage et calcul de la vraisemblance [vrais.m]

- A partir du vecteur input `x` de la fonction, définir les probabilités de transition p_{11} et p_{22} et les sous vecteurs de paramètres suivants :

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \text{ phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \text{ et } \text{VAR_L} = \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}$$

Remarque : les probabilités de transition sont comprises entre 0 et 1. Pour éviter de contraindre l'espace de recherche des paramètres, on appliquera par exemple la fonction `normcdf` (fonction de répartition de la loi normale) aux deux premiers éléments de `x` tels qu'ils appartiennent à $[0,1]$.

- En utilisant les probabilités de transition p_{11} et p_{22} , construire la matrice de transition :

$$PR_TR = \begin{pmatrix} p_{11} & 1 - p_{22} \\ 1 - p_{11} & p_{22} \end{pmatrix}$$

- Construire le vecteur PR_TRF en empilant les deux colonnes de PR_TR.
- Construire les probabilités ergodiques $\pi_1 = P(S_t = 1)$ et $\pi_2 = P(S_t = 2)$ (dans le cas de deux états, on a donné les formules en cours).

Remarque : dans le cas général où l'on a N états, le vecteur des probabilités ergodiques est obtenu comme suit : $\Pi = (A'A)^{-1}A'e_{N+1}$ avec $A = \begin{bmatrix} I_N - P \\ 1' \end{bmatrix}$ et $e_{N+1} = [0 \ 0 \ \dots \ 1]'$

- A partir de PROB__T, construire le vecteur $PROB_ = \begin{pmatrix} P(S_t = 1) \\ P(S_t = 1) \\ P(S_t = 2) \\ P(S_t = 2) \end{pmatrix}$ qui servira à la première

itération du filtre.

- Définir T la taille de l'échantillon. *Indications : on utilisera la taille de endog (fonction length).*
- Définir LIKV = 0 un scalaire dans lequel on stockera la log-vraisemblance (son opposé).
- Définir PR_STL1 et PR_STT1 deux vecteurs de taille T.

- Pour $t=2$ à T, on exécutera ensuite les 5 étapes du filtre et l'on construira progressivement la vraisemblance du modèle (en fait, son opposé car fminunc calcule le minimum de la fonction).

A chaque itération, on stockera dans les vecteurs PR_STL1 et dans PR_STT1 les probabilités $P(S_t = 1|I_{t-1}; \theta)$ et $P(S_t = 2|I_{t-1}; \theta)$, $t = 1, \dots, T$ qui seront utiles dans l'algorithme de lissage.

- *Etape 1* : Stocker dans un vecteur de dimension 4, PROB_DD, les probabilités jointes :

$$PROB_DD_{4 \times 1} = \begin{pmatrix} P(S_t = 1, S_{t-1} = 1 | I_{t-1}; \theta) \\ P(S_t = 2, S_{t-1} = 1 | I_{t-1}; \theta) \\ P(S_t = 1, S_{t-1} = 2 | I_{t-1}; \theta) \\ P(S_t = 2, S_{t-1} = 2 | I_{t-1}; \theta) \end{pmatrix} = \begin{pmatrix} P(S_t = 1 | S_{t-1} = 1) \times P(S_{t-1} = 1 | I_{t-1}; \theta) \\ P(S_t = 2 | S_{t-1} = 1) \times P(S_{t-1} = 1 | I_{t-1}; \theta) \\ P(S_t = 1 | S_{t-1} = 2) \times P(S_{t-1} = 2 | I_{t-1}; \theta) \\ P(S_t = 2 | S_{t-1} = 2) \times P(S_{t-1} = 2 | I_{t-1}; \theta) \end{pmatrix}$$

- *Etape 2* : Stocker ensuite dans un vecteur de dimension 2, PROB_DD, les probabilités prévues :

$$PROB_DD_{2 \times 1} = \begin{pmatrix} P(S_t = 1 | I_{t-1}; \theta) \\ P(S_t = 2 | I_{t-1}; \theta) \end{pmatrix} = \begin{pmatrix} P(S_t = 1, S_{t-1} = 1 | I_{t-1}; \theta) \\ P(S_t = 2, S_{t-1} = 1 | I_{t-1}; \theta) \end{pmatrix} + \begin{pmatrix} P(S_t = 1, S_{t-1} = 2 | I_{t-1}; \theta) \\ P(S_t = 2, S_{t-1} = 2 | I_{t-1}; \theta) \end{pmatrix}$$

- *Etape 3* : Stocker dans le vecteur F_CAST les résidus dans chacun des régimes et calculer la densité jointe $f(Y_t, S_t = i | I_{t-1}; \theta)$ que l'on stockera dans le vecteur de dimension 2, PR_VL :

$$F_CAST = \begin{pmatrix} e_t | S_t = 1 \\ e_t | S_t = 2 \end{pmatrix} = \begin{pmatrix} Y_t - c_1 - \phi_1 Y_{t-1} \\ Y_t - c_2 - \phi_2 Y_{t-1} \end{pmatrix}$$

$$PR_VL_{2 \times 1} = \begin{pmatrix} f(Y_t, S_t = 1 | I_{t-1}; \theta) \\ f(Y_t, S_t = 2 | I_{t-1}; \theta) \end{pmatrix} = \begin{pmatrix} f(Y_t | S_t = 1, I_{t-1}; \theta) \times P(S_t = 1 | I_{t-1}; \theta) \\ f(Y_t | S_t = 2, I_{t-1}; \theta) \times P(S_t = 2 | I_{t-1}; \theta) \end{pmatrix}$$

avec sous l'hypothèse de normalité des résidus :

$$f(Y_t|S_t = i, I_{t-1}; \Theta) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{(Y_t - c_i - \phi_i Y_{t-1})^2}{2\sigma_i^2}\right\}, \quad i = 1, 2.$$

- *Etape 4* : Stocker ensuite dans le vecteur PR_VAL la densité conditionnelle $f(Y_t|I_{t-1}; \Theta)$. Pour cela, on sommer les éléments de PR_VL : $f(Y_t|I_{t-1}; \Theta) = \sum_{i=1}^2 f(Y_t, S_t = i|I_{t-1}; \Theta)$. On stockera par ailleurs dans LIK l'opposé de la log-vraisemblance conditionnelle de Y_t . On sommer ces log-vraisemblances conditionnelles au cours des itérations dans LIKV. La log-vraisemblance totale LIKV sera l'output de la fonction VRAIS.

- *Etape 5* : On calculera enfin le vecteur des probabilités filtrées $P(S_t = i|I_t; \Theta)$ $i = \{1, 2\}$ (utilisée à l'itération suivante du filtre) :

$$PROB_T = \begin{pmatrix} P(S_t = 1|I_t; \Theta) \\ P(S_t = 2|I_t; \Theta) \end{pmatrix} = \begin{pmatrix} f(Y_t, S_t = 1|I_{t-1}; \Theta)/f(Y_t|I_{t-1}; \Theta) \\ f(Y_t, S_t = 2|I_{t-1}; \Theta)/f(Y_t|I_{t-1}; \Theta) \end{pmatrix}$$

- Pour l'itération suivante, on stockera enfin dans un vecteur PROB__ de dim 4 :

$$PROB_ = \begin{pmatrix} P(S_t = 1|I_t; \Theta) \\ P(S_t = 1|I_t; \Theta) \\ P(S_t = 2|I_t; \Theta) \\ P(S_t = 2|I_t; \Theta) \end{pmatrix}$$

III. Algorithme de lissage [lissage.m]

On rappelle que l'algorithme de lissage itère pour $t=T-1$ à 1 le calcul suivant :

$$P(S_t = i|I_T; \theta) = \sum_{j=1}^2 \frac{P(S_{t+1} = j|S_t = i) \times P(S_t = i|Y_t, \dots, Y_1; \theta)}{P(S_{t+1} = j|Y_t, \dots, Y_1; \theta)} \times P(S_{t+1} = j|Y_T, \dots, Y_1; \theta)$$

et que l'algorithme est initialisé par le produit de la dernière étape du filtre $P(S_t = i|Y_T, \dots, Y_1; \Theta)$. On calculera ici $P(S_t = 1|I_T; \theta)$ (la probabilité $P(S_t = 2|I_T; \theta)$ étant bien sûr obtenue par $1 - P(S_t = 1|I_T; \theta)$)

- Recopier à la fin du fichier *lissage.m* la fonction vrais qui a été construite dans la partie II de l'exercice, renommer cette fonction *filtrage* et mettre comme argument les vecteurs PR_STL1 et PR_STT1.

- Sous la fonction lissage

- Stocker dans p11 la probabilité de transition à partir du vecteur x.
- Appeler la fonction filtrage et stocker dans PF et PP les probabilités $P(S_t = 1|I_t; \theta)$ et $P(S_t = 2|I_{t-1}; \theta)$, $t = 1, \dots, T$ qui seront utiles dans l'algorithme de lissage
- Définir T le nombre d'observations de endog (utiliser la fonction size)
- Définir un vecteur PL de taille T en utilisant la fonction zeros. Ce vecteur contiendra les probabilités. Définir le Tème élément de ce vecteur à partir du vecteur PF.
- Itérer pour $t=T-1$ à 1 les calculs suivants :
 - Stocker dans PLI1 $P(S_t = 1, S_{t+1} = 1|I_T; \theta) = \frac{p_{11} \times P(S_t = 1|I_t; \theta)}{P(S_{t+1} = 1|I_t; \theta)} \times P(S_{t+1} = 1|I_T; \theta)$
 - Stocker dans PLI2 $P(S_t = 1, S_{t+1} = 2|I_T; \theta) = \frac{p_{12} \times P(S_t = 1|I_t; \theta)}{P(S_{t+1} = 2|I_t; \theta)} \times P(S_{t+1} = 2|I_T; \theta)$
 - $PL(t) = PLI1 + PLI2$

Facultatif – Tests de diagnostic

Créer une fonction *resid_gen.m* calculant les résidus généralisés du modèle aux instants $t = 2, \dots, T$. Indication : les résidus généralisés du MS-AR(1,1) considéré sont obtenus pour $t = 2, \dots, T$ suivant

$$e_t = \sum_{i=1}^2 \left(\frac{\varepsilon_t | S_t = i}{\sigma_i} \right) P(S_t = i | I_{t-1}; \theta) = \sum_{i=1}^2 \left(\frac{y_t - c_i - \phi_i y_{t-1}}{\sigma_i} \right) P(S_t = i | I_{t-1}; \theta)$$

Appeler la fonction *resid_gen* dans le programme principal *pgmpal* en le vecteur de paramètres estimés et appliquer les fonctions *lbqtest* et *jbtest* afin de vérifier l'absence d'autocorrélation, d'effets ARCH et la normalité des résidus.

Facultatif – Application réelle : Adapter ce programme pour repérer les dates de récession et d'expansion aux Etats-Unis (Hamilton, 1989).

On estimera le processus AR(4) avec saut de la constante suivant :

$$y_t = c_{S_t} + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \phi_4 y_{t-4} + \sigma \varepsilon_t$$

sur le taux de croissance trimestriel américain fourni dans la base GDP_US.xlsx. On comparera les dates de récession obtenues aux dates fournies par le NBER.

Annexe – Simulation d'une chaîne de Markov d'ordre 1

On souhaite simuler un processus (S_t) , $t = 1, \dots, T$ suivant une chaîne de Markov d'ordre un à deux états et définie par la matrice de transition suivante :

$$P = \begin{pmatrix} p_{11} & 1 - p_{22} \\ 1 - p_{11} & p_{22} \end{pmatrix} = \begin{pmatrix} 0,9 & 0,2 \\ 0,1 & 0,8 \end{pmatrix}$$

Méthode: on va utiliser des tirages aléatoires d'une variable aléatoire U distribuée suivant une loi uniforme continue sur $[0,1]$. La fonction de répartition d'une telle variable est donnée par :

$$F(u) = P(U \leq u) = \begin{cases} 0 & \text{si } u < 0 \\ u & \text{si } 0 \leq u < 1. \\ 1 & \text{si } u \geq 1 \end{cases}$$

Ainsi, une variable aléatoire continue distribuée suivant une loi uniforme sur $[0,1]$ a une probabilité de 0,9 de se trouver sous $u=0,9$ et une probabilité 0,1 de se trouver au-dessus. La variable S_t sera alors construite de la façon suivante : On tirera u une réalisation de $U \sim \mathbb{U}_{[0,1]}$ et :

- si $S_{t-1} = 1$ et si $u \leq p_{11} = 0,9$, $S_t = 1$
- si $S_{t-1} = 1$ et si $u > p_{11} = 0,9$, $S_t = 2$
- si $S_{t-1} = 2$ et si $u \leq p_{22} = 0,8$, $S_t = 2$
- si $S_{t-1} = 2$ et si $u > p_{22} = 0,8$, $S_t = 1$

Sous Matlab, on procèdera donc de la façon suivante :

- Créer avec la fonction *zeros* de Matlab un vecteur s de taille T qui contiendra les réalisations de la chaîne. On supposera que son premier élément est égal à un.

- Pour $t=2$ à T , répéter les opérations suivantes :

- Simuler une variable U i. i. d. $U_{[0,1]}$. On utilisera la fonction *rand* de Matlab.
- Remplir le vecteur s de 1 ou de 2 suivant la réalisation i de s à l'itération précédente et suivant que la réalisation u de U est supérieure ou inférieure à p_{ii} .

TP4 : Apprentissage supervisé sous Matlab

L'**objectif** de ce TP est de montrer le potentiel de certaines techniques du *machine learning* pour la prévision de court terme de variables macroéconomiques (ici le taux de croissance du PIB français). On utilisera d'abord des arbres de régression puis des forêts aléatoires, ces dernières permettant de gérer l'instabilité des arbres de régression.

Données : Data_TP4.xlsx

- pib : Taux de croissance du PIB français (source : Insee) ;
- ipi : taux de croissance de la production manufacturière (source : Insee)

L'indice de production industrielle a été converti en fréquence trimestrielle par moyenne des valeurs mensuelles sur chaque trimestre. Nous reportons son taux de croissance trimestriel.

Ouvrir un M-file dans lequel on stockera nos lignes de commandes

File \ New \ Script → Save as TP4.m

Q1. Importer les données de croissance du PIB (pib) et le taux de croissance trimestrielle de l'indice de production manufacturière (ipi). On notera Y la variable prévue, X la variable explicative et n le nombre d'observations. Les modèles seront estimés sur les 100 premières observations et testés sur les 15 dernières. Le vecteur `ind_estim` contiendra les indices des 100 premières observations et le vecteur `ind_test` les indices des 15 dernières observations.

ARBRE DE REGRESSION (REGRESSION TREE)

Q2. Partitionner les 100 premières observations de la base de données de façon aléatoire en un *training set* (50% des observations) et un *validating set* (50% des observations).

Indication : on créera deux vecteurs booléens `ind_train` et `ind_val` de dimension 100 et donnant respectivement les indices des observations du *training set* et de l'*evaluation set*. On répartira les 100 observations de façon aléatoire dans les deux sous-ensembles en utilisant la commande `randsample(100,round(0.5*100))`.

Q3. Utiliser les commandes suivantes pour construire et visualiser l'arbre de régression RT sur le *training set*.

```
RT = fitrtree(X(ind_train),Y(ind_train));  
view(RT, 'Mode', 'graph')
```

Remarque : on peut utiliser les options `MaxNumSplits`, `MinLeafSize`, ou `MinParentSize` pour contrôler le nombre maximum de nœuds de l'arbre, le nombre minimum d'observations dans les feuilles ou dans les nœuds parents. Par exemple, `Mdl = fitrtree(X,Y,'MaxNumSplits',3)` limite à 3 le nombre de nœuds de l'arbre.

Q4. Calculer le nombre maximum d'élagages possibles en utilisant la commande :

```
m = max(RT.PruneList)
```

Remarque : Vous pouvez visualiser chaque niveau sur le graphique précédent .

Pruning level:	0 of 9
----------------	--------

Q5. Pour les niveaux d'élitage `pruneLevels = 0:1:m`, prévoir Y sur le *validation set* :

```
pruneLevels = 0:1:m;  
Yfit = predict(RT,X(ind_val),'SubTrees',pruneLevels)
```

Yfit est une matrice de dimension 50 x (m+1) contenant les prévisions des 50 observations de l'évaluation set pour chaque niveau de profondeur de l'arbre.

Q6. Calculer les mse associées à chaque profondeur pour élaguer l'arbre de façon optimale :

```
mse = mean((Y(ind_val) - Yfit).^2)
```

Remarque : on utilisera de façon équivalente la commande matlab :

```
L = loss(RT,X(ind_val),Y(ind_val),'Subtrees',pruneLevels)
```

Q7. Pour élaguer éventuellement l'arbre RT en fonction des résultats de Q6, utiliser la commande :

```
RT_final = prune(RT,'Level',a);  
view(RT_final,'Mode','graph')  
Yfit_RT = predict(RT_final,X(ind_test));
```

avec **a** le niveau d'élagage souhaité d'après la question 6 (0 : pas d'élagage à `max(RT.PruneList)`).
On stocke dans `Yfit_RT` les prévisions avec l'arbre finalement retenu *sur la plage de test*.

Q8. **Changer la composition du training set**. L'arbre de régression est-il modifié ? Quel inconvénient des arbres de régression met-on ici en évidence ?

FORÊTS ALEATOIRES (RANDOM FOREST)

Q9. En utilisant la commande `TreeBagger`, construire une **forêt aléatoire**. La forêt aléatoire sera constituée de 100 arbres de régression estimés sur des échantillons constitués par tirage au hasard avec remise de 50% des observations de la base et à partir de 1/3 des variables explicatives de la base (ici la seule variable explicative disponible). La taille minimale des feuilles sera fixée à 5 observations. On représentera le mse calculé sur la moitié des observations écartées en fonction du nombre d'arbres de régression dans la forêt.

```
RF = TreeBagger(100,X(1:100),Y(1:100),'Method','regression','InBagFraction',0.5,...  
               'OOBPred','On','MinLeafSize',5);  
plot(oobError(RF))
```

Principales options (voir l'aide en ligne pour une présentation exhaustive) :

- 'Method': 'regression (par défaut, classification) ;
- 'InBagFraction': la fraction d'observations tirées avec remise et utilisée dans le training set pour la construction de chaque arbre (par défaut, 1) ;
- 'NumPredictorsToSample': nombre entier donnant le nombre de prédicteurs retenus pour la construction des arbres (si 'all', bagging, sinon, random forest) ; par défaut, 1/3 des prédicteurs sont tirés au hasard parmi les p.
- 'MinLeafSize': taille minimale des feuilles
- 'OOBPred': 'on' pour conserver l'information sur quelles observations sont écartées (*out of bag*) dans l'évaluation set pour chaque arbre.

Remarque : 1) `b.OOBIndices` permet de visualiser quel échantillon est retenu pour construire chacun des 100 arbres ; 2) on pourra utiliser `view(b.Trees{i},'Mode','graph')` avec `i=1,...,100` pour visualiser un à un les 100 arbres de la forêt.

Q10. Refaire l'exercice pour une taille minimale des feuilles fixée à 10, puis à 15 observations. Quel est le choix optimal ? Cette forêt optimale sera appelée `RF_final`.

```
RF_final = TreeBagger(100,X(1:100),Y(1:100),'Method','regression','InBagFraction',0.5,...  
                  'MinLeafSize',b);
```

COMPARAISON DES DEUX APPROCHES

Q11. Utiliser la forêt aléatoire retenue pour prévoir le taux de croissance du PIB sur l'*evaluation set* utilisé dans la construction des arbres de régression (on utilise ici cet ensemble à des fins de comparaison avec la première méthode).

```
Yfit_RF = predict(RF_final,X(ind_test));
```

Q12. Comparer les performances prévisionnelles des deux méthodes :

```
mse_RT = mean((Y(ind_test) - Yfit_RT).^2)  
mse_RF = mean((Y(ind_test) - Yfit_RF).^2)
```