Stage d'application (2A)

Real-time detection of turning points with linear filters

Alain Quartier-la-Tente
Maître de stage : Jean PALATE (NBB)
07/01/2021
Ensae — 2019-2020

Contexte

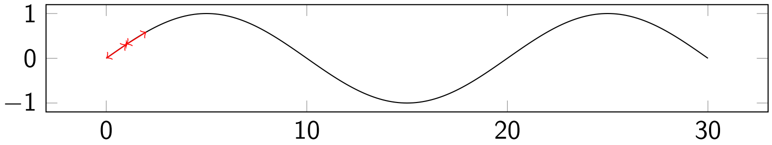
- Stage de 10 semaines effectué dans la cellule Recherche et Développement de la Banque Nationale de Belgique auprès de Jean PALATE (NBB)

- Stage co-dirigé par Dominique LADIRAY (Insee)

- **Objectif** : préparer une thèse sur l'utilisation des filtres asymétriques pour la détection des points de retournement
  - Première revue de la bibliographie
  - Prise en main de quelques programmes R (`rjdfilters`, https://github.com/palatej/rjdfilters)

Introduction (1/2)

- Dans l'analyse de la conjoncture, la détection rapide des points de retournement est très importante

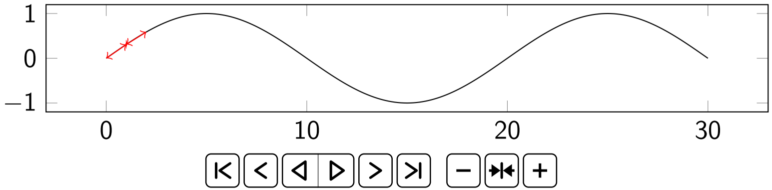


Introduction (1/2)

- Dans l'analyse de la conjoncture, la détection rapide des points de retournement est très importante

- Une série X_t se décompose en plusieurs composantes inobservées :

$$X_t = T_t + C_t + S_t + I_t$$



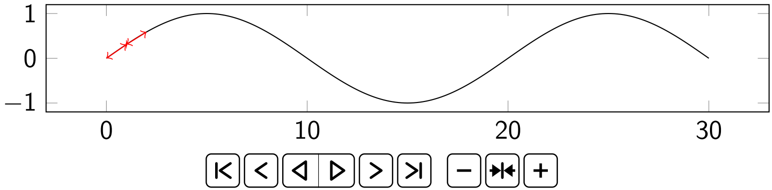
Introduction (1/2)

- Dans l'analyse de la conjoncture, la détection rapide des points de retournement est très importante

- Une série X_t se décompose en plusieurs composantes inobservées :

$$X_t = T_t + C_t + S_t + I_t$$

➔ Méthodes d'extraction de **tendance**/**cycle** (C_t ou $T_t + C_t$) liées aux méthodes de désaisonnalisation



Introduction (2/2)

Moyennes mobiles (ou *filtres linéaires*) omniprésentes dans ces méthodes (ex : X-12ARIMA)
Mathématiquement, à la date t on calcule :

$$M_\theta(X_t) = \sum_{k=-p}^{+f} \theta_k X_{t+k}$$

Introduction (2/2)

Moyennes mobiles (ou *filtres linéaires*) omniprésentes dans ces méthodes (ex : X-12ARIMA)
Mathématiquement, à la date t on calcule :

$$M_\theta(X_t) = \sum_{k=-p}^{+f} \theta_k X_{t+k}$$

- Généralement utilisation de moyennes *symétriques* ($p = f$ et $\theta_{-i} = \theta_i$)
- En fin de période, utilisation de méthodes *asymétriques* : révisions

Introduction (2/2)

Moyennes mobiles (ou *filtres linéaires*) omniprésentes dans ces méthodes (ex : X-12ARIMA)
Mathématiquement, à la date t on calcule :

$$M_\theta(X_t) = \sum_{k=-p}^{+f} \theta_k X_{t+k}$$

- Généralement utilisation de moyennes *symétriques* ($p = f$ et $\theta_{-i} = \theta_i$)
- En fin de période, utilisation de méthodes *asymétriques* : révisions
- Étude des méthodes non-paramétriques qui peuvent s'intégrer dans X-12-ARIMA et dans le logiciel de désaisonnalisation JDemetra+ (maintenu par la NBB).

Sommaire

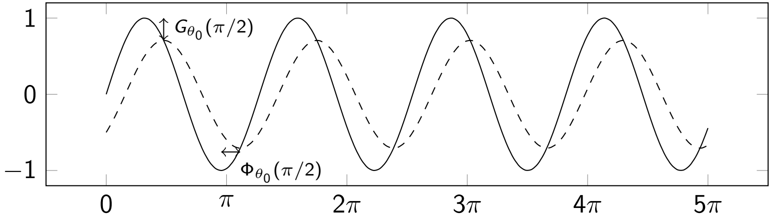
Gain et déphasage

Appliquer M_θ sur $X_t = \sin(\omega t)$ va avoir deux effets :

1. Multiplier le niveau par $G_\theta(\omega)$ (*gain*)
2. Créer un *déphasage* $\Phi_\theta(\omega)/\omega$: affecte détection des points de retournement

Exemple :

$$M_{\theta_0} X_t = \frac{1}{2} X_{t-1} + \frac{1}{2} X_t,\ \omega = \pi/2$$



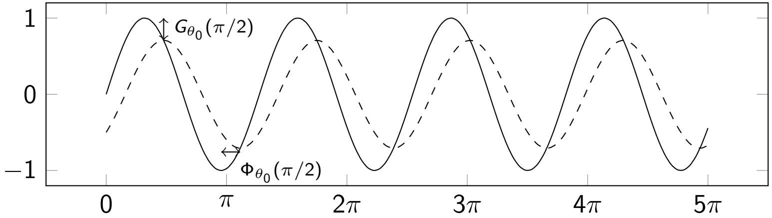
Gain et déphasage

Appliquer M_θ sur $X_t = \sin(\omega t)$ va avoir deux effets :

1. Multiplier le niveau par $G_\theta(\omega)$ (*gain*)
2. Créer un *déphasage* $\Phi_\theta(\omega)/\omega$: affecte détection des points de retournement

Exemple :

$M_{\theta_0}X_t = \frac{1}{2}X_{t-1} + \frac{1}{2}X_t$, $\omega = \pi/2$



➔ Déphasage nul pour MM symétriques

Conservation des tendances polynomiales

$$X_t = TC_t + \varepsilon_t \implies M_\theta X_t = M_\theta TC_t + M_\theta \varepsilon_t$$

Généralement tendance approchée par une tendance polynomiale

➔ Contraintes supplémentaires pour les conserver.

Conservation des tendances polynomiales

$$X_t = TC_t + \varepsilon_t \implies M_\theta X_t = M_\theta TC_t + M_\theta \varepsilon_t$$

Généralement tendance approchée par une tendance polynomiale

➔ Contraintes supplémentaires pour les conserver.
Exemple :

- M_θ conserve les constantes si :

$$M_\theta(a) = a \implies \sum_{k=-p}^{+f} \theta_k = 1$$

Autres indicateurs

1. Réduction de la variance (*Fidelity*) :

$$\mathbb{V}\left[M_\theta \varepsilon_t\right] = \mathbb{V}\left[\varepsilon_t\right] \underbrace{\sum_{k=-p}^{+f} \theta_k^2}_{=F_g}$$

2. Lissage (*Smoothness*) défini par Henderson :

$$S_G = \sum_j (\nabla^3 \theta_j)^2$$

3. Temporalité (*Timeliness*) qui mesure le déphasage dans les fréquences liées au cycle :

$$T_g = \int_0^{2\pi/12} \rho_\theta(\omega) \sin(\varphi_\theta(\omega))^2 \,\mathrm{d}\omega$$

4. D'autres critères définis par Wildi et McElroy (2019) en décomposant l'erreur quadratique moyenne des révisions : A_w, T_w, S_w, R_w.

Sommaire

Approche générale, Proietti and Luati (2008)

On suppose que l'on peut décomposer notre série $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t est localement approché par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket \ : \ y_{t+j} = m_{t+j} + \varepsilon_{t+j}, \quad m_{t+j} = \sum_{i=0}^{d} \beta_i j^i$$

Approche générale, Proietti and Luati (2008)

On suppose que l'on peut décomposer notre série $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t est localement approché par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket \ : \ y_{t+j} = m_{t+j} + \varepsilon_{t+j}, \quad m_{t+j} = \sum_{i=0}^{d} \beta_i j^i$$

Estimation par moindres carrés pondérés (WLS) en minimisant :

$$S(\hat{\beta}_0, \dots, \hat{\beta}_d) = \sum_{j=-h}^{h} \kappa_j (y_{t+j} - \hat{\beta}_0 - \hat{\beta}_1 j - \dots - \hat{\beta}_d j^d)^2$$

κ_j sont les *noyaux*.

Approche générale, Proietti and Luati (2008)

On suppose que l'on peut décomposer notre série $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t est localement approché par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket \ : \ y_{t+j} = m_{t+j} + \varepsilon_{t+j}, \quad m_{t+j} = \sum_{i=0}^{d} \beta_i j^i$$

Estimation par moindres carrés pondérés (WLS) en minimisant :

$$S(\hat{\beta}_0, \dots, \hat{\beta}_d) = \sum_{j=-h}^{h} \kappa_j (y_{t+j} - \hat{\beta}_0 - \hat{\beta}_1 j - \dots - \hat{\beta}_d j^d)^2$$

κ_j sont les *noyaux*.

L'estimateur est $\hat{\beta} = (X'KX)^1 X'Ky$ et :

$$\hat{m}_t = \hat{\beta}_0 = w'y = \sum_{j=-h}^{h} w_j y_{t-j}$$ ➔ estimation par filtre symétrique.

Construction des filtres asymétriques

Plusieurs solutions :

1. Même méthode avec moins de points (DAF) : **sans biais** mais beaucoup de variance

Construction des filtres asymétriques

Plusieurs solutions :

1. Même méthode avec moins de points (DAF) : **sans biais** mais beaucoup de variance
2. Compromis biais-variance : minimisation des révisions sous contraintes plus faibles
   Filtres v dépend d'un paramètre à définir par l'utilisateur
   2.1 *Linear-Constant* (LC) : y_t linéaire et v conserve constante
   2.2 *Quadratic-Linear* (QL) : y_t degré 2 et v conserve polynôme degré 1
   2.3 *Cubic-Quadratic* (CQ) : y_t degré 3 et v conserve polynôme degré 2

Construction des filtres asymétriques

Plusieurs solutions :

1. Même méthode avec moins de points (DAF) : **sans biais** mais beaucoup de variance
2. Compromis biais-variance : minimisation des révisions sous contraintes plus faibles
   Filtres v dépend d'un paramètre à définir par l'utilisateur
   2.1 *Linear-Constant* (LC) : y_t linéaire et v conserve constante
   2.2 *Quadratic-Linear* (QL) : y_t degré 2 et v conserve polynôme degré 1
   2.3 *Cubic-Quadratic* (CQ) : y_t degré 3 et v conserve polynôme degré 2

Pour visualiser les différentes méthodes :

https://aqlt.shinyapps.io/FiltersProperties/

Premières analyses

- Méthode **simple** qui permet de retrouver les principales moyennes mobiles (Henderson, Musgrave, Loess, etc.).

Premières analyses

- ☺ Méthode **simple** qui permet de retrouver les principales moyennes mobiles (Henderson, Musgrave, Loess, etc.).
- ☹ **Déphasage non contrôlé** avec parfois des résultats étonnants

Premières analyses

- ☺ Méthode **simple** qui permet de retrouver les principales moyennes mobiles (Henderson, Musgrave, Loess, etc.).
- ☹ **Déphasage non contrôlé** avec parfois des résultats étonnants $\rightarrow$ possibilité de changer le programme pour le contrôler

Premières analyses

- 🙂 Méthode **simple** qui permet de retrouver les principales moyennes mobiles (Henderson, Musgrave, Loess, etc.).
- 🙁 **Déphasage non contrôlé** avec parfois des résultats étonnants $\rightarrow$ possibilité de changer le programme pour le contrôler

- Privilégier LC ou QL qui réduisent les révisions avec un ajout limité du déphasage.

Premières analyses

- 🙂 Méthode **simple** qui permet de retrouver les principales moyennes mobiles (Henderson, Musgrave, Loess, etc.).
- 🙁 **Déphasage non contrôlé** avec parfois des résultats étonnants $\rightarrow$ possibilité de changer le programme pour le contrôler

- Privilégier LC ou QL qui réduisent les révisions avec un ajout limité du déphasage. ➔ se concentrer sur les filtres asymétriques qui preservent les polynômes de degré au plus 1

L'approche FST, Grun-Rehomme et al (2018)

Minimisation sous contrainte d'une somme pondérée de 3 critères :

$$\begin{cases} \min\limits_{\theta} & J(\theta) = \alpha F_g(\theta) + \beta S_g(\theta) + \gamma T_g(\theta) \\ s.c. & C\theta = a \end{cases}$$

L'approche FST, Grun-Rehomme et al (2018)

Minimisation sous contrainte d'une somme pondérée de 3 critères :

$$\begin{cases} \min\limits_{\theta} & J(\theta) = \alpha F_g(\theta) + \beta S_g(\theta) + \gamma T_g(\theta) \\ s.c. & C\theta = a \end{cases}$$

- Avantages :
  - Solution unique connue analytiquement
  - Filtres asymétriques indépendants des données et du filtre symétrique
  - Outil utile pour évaluer les autres méthodes
- Inconvénient :
  - Poids non normalisés

Filtres RKHS : Dagum and Bianconcini (2008)

- Utilisation de la théorie des RKHS pour approximer le filtre de Henderson
- Avec K_p la **fonction de noyau**, le filtre symétrique :

$$\forall j \in \llbracket -h, h \rrbracket \ : \ w_j = \frac{K_p(j/b)}{\sum_{i=-h}^{h} K_p(i/b)}$$

Avec $b = h + 1$ on retrouve les filtres polynomiaux locaux

Filtres RKHS : Dagum and Bianconcini (2008)

- Utilisation de la théorie des RKHS pour approximer le filtre de Henderson
- Avec K_p la **fonction de noyau**, le filtre symétrique :

$$\forall j \in [\![-h, h]\!] \;:\; w_j = \frac{K_p(j/b)}{\sum_{i=-h}^{h} K_p(i/b)}$$

  Avec $b = h + 1$ on retrouve les filtres polynomiaux locaux

- Pour les filtres asymétriques :

$$\forall j \in [\![-h, q]\!] \;:\; w_{a,j} = \frac{K_p(j/b)}{\sum_{i=-h}^{q} K_p(i/b)}$$

Filtres asymétriques

b déterminé par optimisation en minimisant :

- les révisions : $b_{q,\gamma} = \underset{b_q \in]h;2h+1]}{\min} \sqrt{2\int_0^{\pi} |\Gamma_s(\omega) - \Gamma_\theta(\omega)|^2 \,\mathrm{d}\omega}$
- les révisions liées à la fonction de gain
  $b_{q,G} = \underset{b_q \in]h;2h+1]}{\min} \sqrt{2\int_0^{\pi} \left(\rho_s(\omega) - \rho_\theta(\omega)\right)^2 \,\mathrm{d}\omega}$
- les révisions liées au déphasage
  $b_{q,\varphi} = \underset{b_q \in]h;2h+1]}{\min} 8\int_0^{2\pi/12} \rho_s(\lambda)\rho_\theta(\lambda)\sin^2\left(\frac{\varphi_\theta(\omega)}{2}\right) \,\mathrm{d}\omega$

Avantages et inconvénients

- Avantages :
  - Méthode généralisable à des filtres avec fréquences irrégulières
- Inconvénients :
  - MM ne conservent que les constantes
  - Il peut y avoir des problèmes d'optimisation (plusieurs minimums, etc.)

Sommaire

Comparaison à partir de la méthode FST

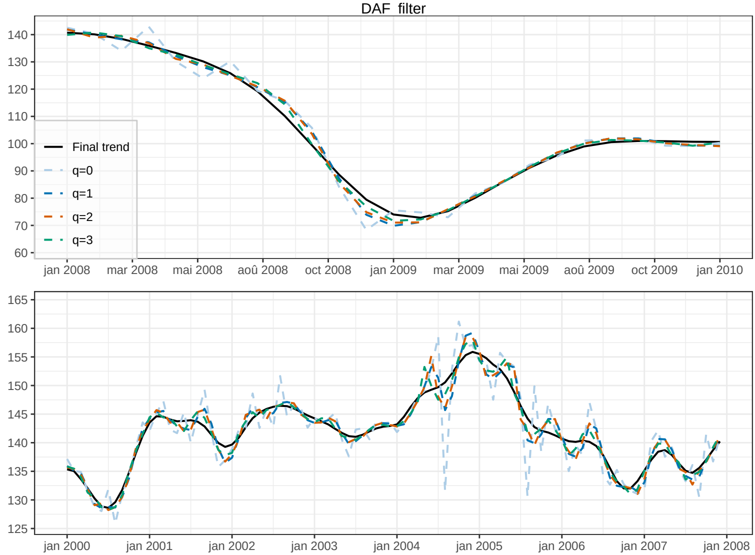
On regarde si on peut faire **mieux sur les trois critères** avec la méthode FST et **mêmes contraintes polynomiales** :
https://aqlt.shinyapps.io/FSTfilters/

Comparaison à partir de la méthode FST

On regarde si on peut faire **mieux sur les trois critères** avec la méthode FST et **mêmes contraintes polynomiales** :
https://aqlt.shinyapps.io/FSTfilters/

- Meilleurs résultats avec FST qu'avec RKHS avec **mêmes contraintes polynomiales**
- Meilleurs résultats avec FST qu'avec LC, sauf lorsque l'on contrôle le déphasage pour $q = 0, 1$
- Pas de meilleur résultat avec FST qu'avec QL pour $q = 0, 1$

Un exemple d'application : IPI-FR CL1 (1/2)



Sommaire

Conclusion

- DAF (utilisé dans STL) donne les moins bons résultats sur détection des points de retournement et révisions
- Minimiser timeliness n'implique pas une détection plus rapide de tous les points de retournement
- Introduire du biais dans la préservation des tendances polynomiales permet d'améliorer la détection rapide des points de retournement

Conclusion

- DAF (utilisé dans STL) donne les moins bons résultats sur détection des points de retournement et révisions
- Minimiser timeliness n'implique pas une détection plus rapide de tous les points de retournement
- Introduire du biais dans la préservation des tendances polynomiales permet d'améliorer la détection rapide des points de retournement
- À partir de $q = 2$ les méthodes donnent des résultats similaires

Ouverture

- Filtres FST permettent de construire des filtres “meilleurs” que les autres

Ouverture

- Filtres FST permettent de construire des filtres “meilleurs” que les autres
- Beaucoup de paramètres : lesquels sont pertinents ?

Ouverture

- Filtres FST permettent de construire des filtres “meilleurs” que les autres
- Beaucoup de paramètres : lesquels sont pertinents ?
- Quid de la longueur des filtres et des séries avec d’autres fréquences ?

Ouverture

- Filtres FST permettent de construire des filtres “meilleurs” que les autres
- Beaucoup de paramètres : lesquels sont pertinents ?
- Quid de la longueur des filtres et des séries avec d’autres fréquences ?
- Quid de l’impact des points atypiques : étude des méthodes robustes

Merci pour votre attention

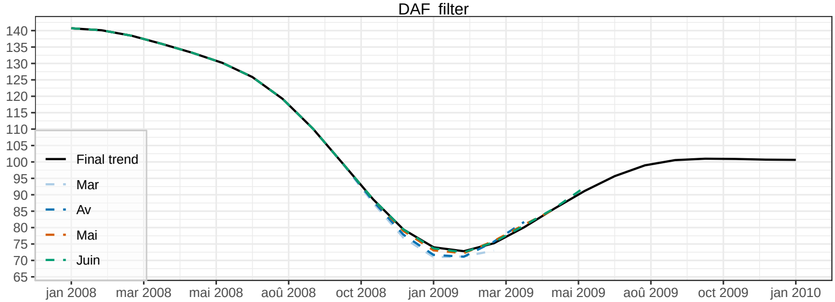
AQLT/AsymmetricFilters

Rapport du projet

Slides

palatej/rjdfilters

Annexe 1


DAF_filter
140
135
130
125
120
115
110
105
100
95
90
85
80
75
70
65
Final trend
Mar
Av
Mai
Juin
jan 2008
mar 2008
mai 2008
aoû 2008
oct 2008
jan 2009
mar 2009
mai 2009
aoû 2009
oct 2009
jan 2010