

Projet de programmation de méthodes mathématiques pour la finance

Jean-Baptiste Durand et Ollivier Taramasco

Ensimag

Filière MeQA

Novembre 2017 – mars 2018

- **Thématique**
- **Objectifs**
- **Organisation des groupes**
- **Évaluation**
- **Planning**
- **Documentation**
- **Cahier des charges**

Thématique

- **Modèles ARMA à résidus GARCH et variantes**

$$Y(t) - \sum_{i=1}^r AR(i)Y(t-i) = C_{ARMA} + u(t) + \sum_{j=1}^m MA(j)u(t-j)$$

$$u(t) = \sqrt{h(t)}\varepsilon(t)$$

$$h(t) = C_{GARCH} + \sum_{k=1}^p ARCH(k)u^2(t-k) + \sum_{l=1}^q GARCH(l)h(t-l)$$

$(\varepsilon(t))_{t \in \mathbb{Z}}$ suite de variables aléatoires indépendantes centrées réduites

- **Simulation**
- **Calcul de la « vraisemblance » et son gradient**
- **Estimation des paramètres**
- **Conception de l'architecture d'un programme**
- **Programmation objet en C++**

stat

info

Objectifs

- **Comprendre par la pratique comment s'utilise un modèle statistique :**
 - Simulation
 - Intérêt de la vraisemblance et son gradient (méthodes numériques, formelles)
 - Estimation, Inférence (Intervalles de confiance, etc.)
- **Calcul d'espérances et variances conditionnelles, vraisemblance, scores**
- **Conception de l'architecture d'un programme :**
 - modularité
 - généricité
 - interfaces
- **Programmation objet en C++ :**
 - classes et méthodes
 - « virtualité »
 - débuggage, compilation
- **Utilisation de bibliothèque scientifique**
- **Documentation du code**

Modèle - rappels

$$Y(t) - \sum_{i=1}^r AR(i)Y(t-i) = C_{ARMA} + u(t) + \sum_{j=1}^m MA(j)u(t-j)$$

$$u(t) = \sqrt{h(t)}\varepsilon(t)$$

$$h(t) = C_{GARCH} + \sum_{k=1}^p ARCH(k)u^2(t-k) + \sum_{l=1}^q GARCH(l)h(t-l)$$

$(\varepsilon(t))_{t \in \mathbb{Z}}$ suite de variables aléatoires indépendantes centrées réduites

partie ARMA /
espérance
conditionnelle

(... explication ?)

partie GARCH /
variance
conditionnelle

- **Série chronologique : temps discret**
- **Observations : $Y(t)$ pour $n \geq t \geq 0$. Les $U(t)$, $h(t)$ et $\varepsilon(t)$ sont inconnus**
- **On peut faire comme si les $Y(t)$ étaient connus pour $t < 0$**
- **Modélisation de la dépendance à travers :**
 - l'espérance conditionnelle (ARMA)
 - la variance conditionnelle (GARCH)
 - caractère additif des équations
- **Résidus : bruit (centré, réduit) spécifiant une famille de modèles (gaussien, Student)**

(Pseudo-) Vraisemblance

- On note θ l'ensemble des paramètres et L_n la vraisemblance :

$$L_n(\theta) = f_\theta(y_1, \dots, y_n)$$

- Pour tout processus, on a $\log(L_n(\theta)) = \sum_{t=1}^n l_t(\theta)$ (... pourquoi ?)

avec $l_t(\theta) = \log(f_\theta(y_t | y_1, \dots, y_{t-1}))$

- L'expression de $l_t(\theta)$ est très compliquée car la définition de Y_t fait intervenir les quantités inconnues $U(t)$, $h(t)$ et $\varepsilon(t)$
- Du coup on calcule la pseudo-vraisemblance, obtenue en remplaçant

$$l_t(\theta) \text{ par } \log(f_\theta(y_t | (y_{t-s}, u_{t-s}, h_{t-s}, \varepsilon_{t-s})_{s>0})) \text{ dans } L_n(\theta)$$

- L'estimateur de θ obtenu en maximisant cette quantité est « en général » convergent, asymptotiquement sans biais et gaussien
- De manière abrégée, on parlera de « vraisemblance » à la place de la pseudo-vraisemblance

Problématiques

- **But ultime du projet :**

- Simulation de modèles ARMA/GARCH avec différentes familles de résidus
- Estimation des paramètres par maximum de vraisemblance
- Intervalles de confiance, tests, sélection de modèles (?)
- Potentielles extensions à d'autres modèles d'espérance / variance conditionnelle

- **Étapes intermédiaires :**

- calcul de l'espérance conditionnelle, la variance conditionnelle, la loi conditionnelle

$$(de \quad y_t | (y_{t-s}, u_{t-s}, h_{t-s}, \varepsilon_{t-s})_{s>0})$$

- calcul de la log-vraisemblance et de son gradient
- optimisation de la log-vraisemblance par une méthode numérique (utilisant la valeur de la log-vraisemblance et de son gradient)

- **Spécialisations et extensions possibles :**

- Stratégies d'optimisation : comparatif
- Création d'une interface web ou Excel
- Implémentation de diverses variantes de modèles
- Tests statistiques

tronc commun

spécialisation

6

Ppmmf

J.-B. Durand & O. Taramasco

Travail par équipes

- **Tronc commun (6 semaines), J.-B. Durand :**
 - travail par équipes de 3 ou 4
 - choix libre des équipes
 - programme de travail hebdomadaire à respecter collectivement
 - évaluation individuelle : TP en temps limité
 - plateforme : Linux à l'école (environnement de développement netbeans)
- **Spécialisation (6 semaines), O. Taramasco :**
 - travail par groupes un peu plus importants
 - coordination nécessaire entre les groupes : liens entre l'estimation de paramètres, les intervalles de confiance et l'interfaçage
 - soutenance à la dernière séance ou juste après
 - plateforme : Windows (Visual Studio)
- **Présence obligatoire sauf entretien, maladie (justificatif)**
- **Gestion des équipes : sous TEIDE (rendus d'équipe)**
- **Gestion des sources par dépôt git**

Évaluation - Livrables

- Une séance de TP individuel en temps limité, noté (pas de connexion réseau, ni d'accès à son compte personnel)
 - sur une partie du travail donné à faire l'une des semaines précédentes
 - code à compléter
 - programme de test exécutable qui sert de référence pour l'évaluation
- Livrables à rendre par trinôme :
 - schéma des classes et méthodes
 - simulation et calcul de vraisemblance
- Soutenance orale par équipe, à l'issue de la phase de spécialisation
- Après chaque livrable rendu par les trinômes, l'équipe enseignante livre des sources de manière à ce que les binômes ne soient pas bloqués pour la suite

**copie strictement
interdite
(impossible ?)**

**copie fortement
déconseillée
(éval. indiv.)**

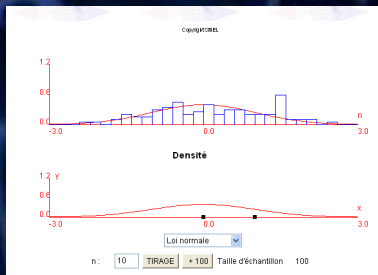
Tâches : tronc commun

- * Architecture du projet : classes, méthodes (diagrammes, pseudo-code)
- * Calculs d'espérances et variances conditionnelles
- * Simulation
- * Calcul de la vraisemblance
- * Calcul du gradient de la log-vraisemblance (numérique, formel) et estimation des paramètres
- * Augmentation graduelle de la complexité :
 - ARMA pur \rightarrow ARMA / GARCH
 - résidus gaussiens \rightarrow résidus gaussiens ou Student
 - ...

Tâches : spécialisation

semaine 6 TP noté individuel (3h)

Semaines 7-12 :



$$\arg \max_{\theta} \sum_{t=1}^n \log(f_{\theta}(y_t | (y_{t-s}, u_{t-s}, h_{t-s}, \varepsilon_{t-s})_{s>0}))$$

$$\sigma_t^2 = c + a[Y_{t-1} - \mu_{t-1}]^2$$
$$\mu_t = \alpha + \beta \sigma_t^2$$

...

* Soutenance par équipe de spécialisation

Ppmmf
J.-B. Durand & O. Taramasco

10

Documentation

- Chamilo

- installation de la bibliothèque scientifique GSL et guide d'utilisation
- récupérer Visual Studio (pas indispensable, plutôt après la semaine 6)
- planning
- directives
- code distribué par les enseignants

- Doxygen

- pour que vous documentiez votre travail
- insérer des balises particulières dans le code C++ pour le documenter

```
/** exemple de commentaire pour une methode dans un header */
```

- documenter son code et ses classes est **obligatoire**

Conventions

- **Vous devez respecter des conventions de nommage :**
 - des fichiers
 - des classes
 - des méthodes
 - des variables
 - des arguments de méthodes
- **Nous vous proposons des conventions de nommage (sur chamilo). En particulier :**
 - un fichier .h et un .cpp par classe
 - noms anglais et évocateurs (myLikelihood plutôt que toto)
 - préfixe the pour les noms d'argument, my pour les variables locales, c pour les classes, m pour les membres ou méthodes (mt protected, mv private)
 - etc.
- **Vous pouvez adopter vos propres conventions, mais vous devez alors les expliciter dans votre documentation (avant toute chose).**

Cahier des charges

- Vos structures de données, architecture logicielle et classes doivent permettre :
 - de représenter des modèles ARMA/GARCH quelconques (y compris AR purs, GARCH purs, etc.)
 - de simuler ces modèles
 - de calculer l'espérance et la variance conditionnelles de Y_t
 - de calculer la log-vraisemblance, son gradient et sa matrice hessienne
 - d'estimer les paramètres de ces modèles
- Pour y arriver, nous recommandons :
 - d'exploiter la décomposition additive de l'espérance conditionnelle
 - la modularité des modèles AR, MA, ARCH, GARCH : ajouter un 5ème modèle doit se traduire par l'ajout d'une classe, sans modification des autres classes
 - la modularité de la loi des résidus : de même que ci-dessus pour l'ajout d'une autre loi
 - de séparer les structures de données et les algorithmes de calcul d'espérance et de variance conditionnelles

Tâches à réaliser dès aujourd'hui

- **Constitution des équipes sous TEIDE**
- **Spécification des algorithmes de simulation, calcul de vraisemblance et son gradient**
 - recenser les quantités : observations, résidus, paramètres, etc.
 - quantités en entrée et en sortie

Tâches à réaliser dès aujourd'hui

- **Constitution des équipes sous TEIDE**
- **Spécification des algorithmes de simulation, calcul de vraisemblance et son gradient**
 - recenser les quantités : observations, résidus, paramètres, etc.
 - quantités en entrée et en sortie
- **Conception des classes :**
 - membres
 - méthodes et leurs arguments

Tâches à réaliser dès aujourd'hui

- **Constitution des équipes sous TEIDE**
- **Spécification des algorithmes de simulation, calcul de vraisemblance et son gradient**
 - recenser les quantités : observations, résidus, paramètres, etc.
 - quantités en entrée et en sortie
- **Conception des classes :**
 - membres
 - méthodes et leurs arguments
- **Définition des modèles et algorithmes (opt. gradient)**
 - en pseudo-code
 - exemple de pseudo-code acceptable

```
C = AR( $\theta_{AR}$ )  
s = 0  
For i=1 to C.order() Do  
    s = s + C.coef(j) * Y.val(t-i)
```


Tâches à réaliser dès aujourd'hui

- **Constitution des équipes sous TEIDE**
- **Spécification des algorithmes de simulation, calcul de vraisemblance et son gradient**
 - recenser les quantités : observations, résidus, paramètres, etc.
 - quantités en entrée et en sortie
- **Conception des classes :**
 - membres
 - méthodes et leurs arguments
- **Définition des modèles et algorithmes (opt. gradient)**
 - en pseudo-code
 - exemple de pseudo-code acceptable
 - contre-exemple

$$\text{calculer } \sum_{i=1}^p AR(i) y_{t-i}$$

le pseudo-code doit mettre en évidence le rôle des structures de données (classes...)

Tâches à réaliser dès aujourd'hui

- **Constitution des équipes sous TEIDE**
- **Spécification des algorithmes de simulation, calcul de vraisemblance et son gradient**
 - recenser les quantités : observations, résidus, paramètres, etc.
 - quantités en entrée et en sortie
- **Conception des classes :**
 - membres
 - méthodes et leurs arguments
- **Définition des modèles et algorithmes (opt. gradient)**
 - en pseudo-code
 - exemple de pseudo-code acceptable
 - contre-exemple
- **Pour des modèles ARMA purs à résidus gaussiens, mais en respectant le cahier des charges (généricité du design)**
- **Recommandations. Sur une feuille de papier :**
 - réaliser formellement les calculs à un niveau de détail adéquat

Tâches à réaliser d'ici 7 jours

- **Rendre sous TEIDE un schéma des classes avec les membres et méthodes**
- **Préparer un exposé oral avec transparents**
- **Expliquer comment on peut faire un calcul de vraisemblance, simuler un modèle, etc.**
- **Montrer en quoi ajouter par exemple une partie « régression » à l'espérance conditionnelle du modèle n'implique de modifier aucune classe existante.**
- **Comment feriez-vous pour dupliquer un modèle déjà existant si vous ne connaissez pas les instructions qui l'ont défini?**