MODE 2004
Le Havre
25-27 mars

Approche variationnelle et numérique de problèmes de commande optimale stochastique

Systems and Optimization Working Group (SOWG)

guy.cohen@mail.enpc.fr

Approche variationnelle et numérique de problèmes de commande optimale stochastique

Systems and Optimization Working Group (SOWG)

Ce groupe est composé de

Pierre Carpentier
(UMA-ENSTA)
Jean-Philippe Chancelier
(CERMICS-ENPC)
Guy Cohen
(CERMICS-ENPC & INRIA)
Michel Cohen de Lara
(CERMICS-ENPC)

guy.cohen@mail.enpc.fr

Approche variationnelle et numérique de problèmes de commande optimale stochastique

Systems and Optimization Working Group (SOWG)

Ce groupe est composé de

Pierre Carpentier
(UMA-ENSTA)

Jean-Philippe Chancelier
(CERMICS-ENPC)

Guy Cohen
(CERMICS-ENPC & INRIA)

Michel Cohen de Lara
(CERMICS-ENPC)

Laetitia Andrieu
(CERMICS-ENPC)

Kengy Barty
(Univ. Guadeloupe & CERMICS-ENPC)

Anes Dallagi
(Univ. Paris I & CERMICS-ENPC)

Cyrille Strugarek
(EDF & CERMICS-ENPC)

1. Introduction

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.

guy.cohen@mail.enpc.fr

1. Introduction

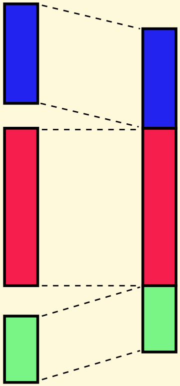
- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible. . .

MODE 2004
Le Havre
25-27 mars

guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible…

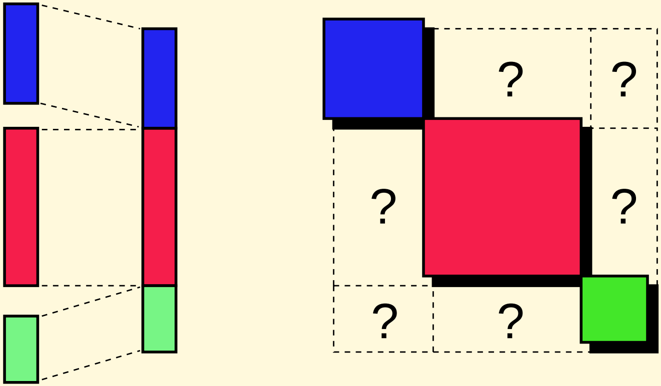


guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.

- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible. . .



guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible. . .

  Dit autrement,

  – en commande optimale déterministe, on décompose l'espace d'*arrivée* de fonctions temps $\to$ espace ;

guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible…

Dit autrement,

- en commande optimale déterministe, on décompose l'espace d'*arrivée* de fonctions temps $\rightarrow$ espace ;
- en commande optimale stochastique, on décompose l'espace de *départ* de fonctions temps $\times$ espace $\rightarrow$ espace ou $\mathbb{R}$;

guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible. . .
- . . . d'où la nécessité d'un point de vue *variationnel* sur ces problèmes. . .

guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible…
- …d'où la nécessité d'un point de vue *variationnel* sur ces problèmes…
- …et aussi d'un point de vue numérique (*“discrétisation”*).

guy.cohen@mail.enpc.fr

1. Introduction

- **Programmation Dynamique stochastique :** la malédiction de la dimension.
- **Programmation Dynamique et méthodes de décomposition-coordination :** le mariage impossible. . .
- . . . d'où la nécessité d'un point de vue *variationnel* sur ces problèmes. . .
- . . . et aussi d'un point de vue numérique (*"discrétisation"*).
- Dans cette tentative, la principale difficulté est le "bon" traitement variationnel et numérique des *contraintes d'information*.

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

$$\min_{u(\cdot)} \mathbb{E} J\big(u(\cdot), \xi(\cdot)\big) \,,$$

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

$$\min_{u(\cdot)} \mathbb{E} J\big(u(\cdot), \xi(\cdot)\big) \; ,$$

  - soumis éventuellement à une contrainte de mesurabilité

    "u est $\mathcal{H}$-mesurable",

    $\mathcal{H}$ étant une sous-σ-algèbre (ou sous-tribu) de $\mathcal{A}$,

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

$$\min_{u(\cdot)} \mathbb{E} J\big(u(\cdot), \xi(\cdot)\big) \; ,$$

  – soumis éventuellement à une contrainte de mesurabilité

  “u est $\mathcal{H}$-mesurable”,

  $\mathcal{H}$ étant une sous-σ-algèbre (ou sous-tribu) de $\mathcal{A}$,

  – $\mathcal{H}$ est en général la σ-algèbre engendrée par une *variable d’observation* $y : \Omega \to \mathcal{Y}$,

Page 3 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

$$\min_{u(\cdot)} \mathbb{E} J\big(u(\cdot), \xi(\cdot)\big) \, ,$$

  - soumis éventuellement à une contrainte de mesurabilité
    "u est $\mathcal{H}$-mesurable",
    $\mathcal{H}$ étant une sous-σ-algèbre (ou sous-tribu) de $\mathcal{A}$,
  - $\mathcal{H}$ est en général la σ-algèbre engendrée par une *variable d'observation* $y : \Omega \to \mathcal{Y}$,
  - ou bien $y = h(u, \omega)$ (alors $\mathcal{H} = \mathcal{H}_u$ i.e. $\mathcal{H}$ dépend de u).

guy.cohen@mail.enpc.fr

MODE 2004
Le Havre
25-27 mars

La classe de problèmes considérée

- espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$,
- espace des commandes $\mathcal{U}$, espace des bruits Ξ,
- fonction coût $J : \mathcal{U} \times \Xi \to \mathbb{R}$ (une intégrande normale),
- bruit $\xi : \Omega \to \Xi$ (une v.a.),
- commande admissible $u : \Omega \to \mathcal{U}$ (une v.a. telle que $J\big(u(\cdot), \xi(\cdot)\big)$ est mesurable et intégrable),

$$\min_{u(\cdot)} \mathbb{E} J\big(u(\cdot), \xi(\cdot)\big) ,$$

  - soumis éventuellement à une contrainte de mesurabilité

    “u est $\mathcal{H}$-mesurable”,

    $\mathcal{H}$ étant une sous-σ-algèbre (ou sous-tribu) de $\mathcal{A}$,
  - $\mathcal{H}$ est en général la σ-algèbre engendrée par une *variable d’observation* $y : \Omega \to \mathcal{Y}$,
  - ou bien $y = h(u, \omega)$ (alors $\mathcal{H} = \mathcal{H}_u$ i.e. $\mathcal{H}$ dépend de u).

Au moins lorsque $\mathcal{H}$ ne dépend pas de u, on note $V(\mathcal{P}, \mathcal{H})$ le *coût optimal*.

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d’optimisation stochastique ;

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *“sans effet dual”* ;

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *"sans effet dual"* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d'information n'est pas triviale ;

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *"sans effet dual"* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d'information n'est pas triviale ;
- quelques résultats théoriques de convergence ;

Plein écran
Zoom Main
Copier Coller
Signets Dest.

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *"sans effet dual"* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d'information n'est pas triviale ;
- quelques résultats théoriques de convergence ;
- ...puis on passe à la pratique : un point de vue algébrique (*treillis des partitions* d'un ensemble)

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d’optimisation stochastique ;
- les problèmes que l’on peut espérer aborder avec un point de vue variationnel sont les problèmes *“sans effet dual”* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d’information n’est pas triviale ;
- quelques résultats théoriques de convergence ;
- …puis on passe à la pratique : un point de vue algébrique (*treillis des partitions* d’un ensemble) combiné à l’idée de *quantification* sont proposés ;

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *"sans effet dual"* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d'information n'est pas triviale ;
- quelques résultats théoriques de convergence ;
- . . . puis on passe à la pratique : un point de vue algébrique (*treillis des partitions* d'un ensemble) combiné à l'idée de *quantification* sont proposés ;
- des résultats expérimentaux sur la *synthèse de feedback* sont examinés ;

guy.cohen@mail.enpc.fr

Dans cet exposé, on discute les points suivants :

- typologie des problèmes d'optimisation stochastique ;
- les problèmes que l'on peut espérer aborder avec un point de vue variationnel sont les problèmes *"sans effet dual"* ;
- même pour ces problèmes, la représentation *discrétisée* des contraintes d'information n'est pas triviale ;
- quelques résultats théoriques de convergence ;
- ...puis on passe à la pratique : un point de vue algébrique (*treillis des partitions* d'un ensemble) combiné à l'idée de *quantification* sont proposés ;
- des résultats expérimentaux sur la *synthèse de feedback* sont examinés ;
- au vu de ces résultats une approche alternative basée sur la résolution de *conditions de type Kuhn-Tucker* est envisagée.

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\varnothing, \Omega\}$, i.e. u est *déterministe*.

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\varnothing, \Omega\}$, i.e. u est *déterministe*. On doit donc résoudre

$$\min_u \mathbb{E} J\big(u, \xi(\omega)\big) \; .$$

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\emptyset, \Omega\}$, i.e. u est *déterministe*. On doit donc résoudre

$$\min_u \mathbb{E} J\big(u, \xi(\omega)\big) \ .$$

- Essentiellement, pour calculer $V(\mathcal{P})$, il nous faut une approximation numérique (*finie*) $\mathcal{P}_n$ de $\mathcal{P}$ (Monte Carlo).

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\emptyset, \Omega\}$, i.e. u est *déterministe*. On doit donc résoudre
$$\min_u \mathbb{E} J\big(u, \xi(\omega)\big) .$$
- Essentiellement, pour calculer $V(\mathcal{P})$, il nous faut une approximation numérique (*finie*) $\mathcal{P}_n$ de $\mathcal{P}$ (Monte Carlo). Abondante littérature sur la convergence de suites $\{\mathcal{P}_n\}$ vers $\mathcal{P}$ (*distance de Fortet-Mourier*, etc.)

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\varnothing, \Omega\}$, i.e. u est *déterministe*. On doit donc résoudre
$$\min_u \mathbb{E} J\big(u, \xi(\omega)\big) .$$
- Essentiellement, pour calculer $V(\mathcal{P})$, il nous faut une approximation numérique (*finie*) $\mathcal{P}_n$ de $\mathcal{P}$ (Monte Carlo). Abondante littérature sur la convergence de suites $\{\mathcal{P}_n\}$ vers $\mathcal{P}$ (*distance de Fortet-Mourier*, etc.) et sur la convergence de $V(\mathcal{P}_n)$ et l'arg-min u (*epi-convergence*).

guy.cohen@mail.enpc.fr

2. Typologie des problèmes

2.1. Optimisation en boucle ouverte (BO)

- C'est le cas où $\mathcal{H}$ est la tribu grossière $\{\varnothing, \Omega\}$, i.e. u est *déterministe*. On doit donc résoudre

$$\min_u \mathbb{E} J\big(u, \xi(\omega)\big) .$$

- Essentiellement, pour calculer $V(\mathcal{P})$, il nous faut une approximation numérique (*finie*) $\mathcal{P}_n$ de $\mathcal{P}$ (Monte Carlo). Abondante littérature sur la convergence de suites $\{\mathcal{P}_n\}$ vers $\mathcal{P}$ (*distance de Fortet-Mourier*, etc.) et sur la convergence de $V(\mathcal{P}_n)$ et l'arg-min u (*epi-convergence*).

- On peut aussi construire l'approximation $\mathcal{P}_n$ de $\mathcal{P}$ et optimiser simultanément (*gradient stochastique*).

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un “feedback”).

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un “feedback”).
- Mais il y a des cas plus subtiles : y dépend de u (ou d'une partie de u) mais la σ-algèbre engendrée n'en dépend pas

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un "feedback").

- Mais il y a des cas plus subtiles : y dépend de u (ou d'une partie de u) mais la σ-algèbre engendrée n'en dépend pas : c'est l'*absence d'effet dual*.

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un “feedback”).
- Résoudre $\quad \min_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big) \quad$ sous $\quad u(\cdot)$ $\mathcal{H}$-mesurable.

Introduction
Typologie des . . .
Un exemple en . . .
Aperçu succinct des . . .
Treillis des partitions . . .
Discrétisation, . . .
Résultats . . .
Conditions . . .

Page 6 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un “feedback”).
- Résoudre $\min_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big)$ sous $u(\cdot)$ $\mathcal{H}$-mesurable.
- Ce problème est équivalent à $\min_{u(\cdot)} \mathbb{E}\Big(G\big(u(\omega), \xi(\omega)\big)\Big)$ avec, pour tout u *déterministe*, $G(u, \cdot) = \mathbb{E}\big(J(u, \cdot) \mid \mathcal{H}\big)$.

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un "feedback").
- Résoudre $\min_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big)$ sous $u(\cdot)$ $\mathcal{H}$-mesurable.
- Ce problème est équivalent à $\min_{u(\cdot)} \mathbb{E}\Big(G\big(u(\omega), \xi(\omega)\big)\Big)$ avec, pour tout u *déterministe*, $G(u, \cdot) = \mathbb{E}\big(J(u, \cdot) \mid \mathcal{H}\big)$.
- Si on approxime $\mathcal{H}$ par la σ-algèbre $\mathcal{H}_k$ engendrée par une *partition finie* $\{\Omega_i\}_{i=1,\dots,k}$, le problème SIS se ramène à un *nombre fini de problèmes en BO*

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un "feedback").
- Résoudre $\min_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big)$ sous $u(\cdot)$ $\mathcal{H}$-mesurable.
- Ce problème est équivalent à $\min_{u(\cdot)} \mathbb{E}\Big(G\big(u(\omega), \xi(\omega)\big)\Big)$ avec, pour tout u *déterministe*, $G(u, \cdot) = \mathbb{E}\big(J(u, \cdot) \mid \mathcal{H}\big)$.
- Si on approxime $\mathcal{H}$ par la σ-algèbre $\mathcal{H}_k$ engendrée par une *partition finie* $\{\Omega_i\}_{i=1,\dots,k}$, le problème SIS se ramène à un *nombre fini de problèmes en BO*
  - $u(\cdot) \sim \{u_i\}_{i=1,\dots,k}$ *constante par morceaux* sur $\{\Omega_i\}$,

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un "feedback").
- Résoudre $\min\limits_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big)$ sous $u(\cdot)$ $\mathcal{H}$-mesurable.
- Ce problème est équivalent à $\min\limits_{u(\cdot)} \mathbb{E}\Big(G\big(u(\omega), \xi(\omega)\big)\Big)$ avec, pour tout u *déterministe*, $G(u, \cdot) = \mathbb{E}\big(J(u, \cdot) \bigm| \mathcal{H}\big)$.
- Si on approxime $\mathcal{H}$ par la σ-algèbre $\mathcal{H}_k$ engendrée par une *partition finie* $\{\Omega_i\}_{i=1,\dots,k}$, le problème SIS se ramène à un *nombre fini de problèmes en BO*
  - $u(\cdot) \sim \{u_i\}_{i=1,\dots,k}$ *constante par morceaux* sur $\{\Omega_i\}$,
  - $V(\mathcal{P}, \mathcal{H}_k) = \sum\limits_{i=1}^{k} \mathcal{P}(\Omega_i) \times \min\limits_{u_i} \mathbb{E}\Big(J\big(u_i, \xi(\omega)\big) \Bigm| \Omega_i\Big)$,

Introduction
Typologie des ...
Un exemple en ...
Aperçu succinct des ...
Treillis des partitions ...
Discrétisation, ...
Résultats ...
Conditions ...
Page 6 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

2.2. Structure d'information statique (SIS)

C'est le cas où $\mathcal{H}$ ne dépend pas de u,

- par exemple quand l'observation $y(\cdot)$ est *entièrement* disponible *avant* de prendre la décision u, donc y ne dépend pas de u et $u(\cdot) = f\big((y(\cdot)\big)$ (f est un "feedback").
- Résoudre $\min_{u(\cdot)} \mathbb{E} J\big(u(\omega), \xi(\omega)\big)$ sous $u(\cdot)$ $\mathcal{H}$-mesurable.
- Ce problème est équivalent à $\min_{u(\cdot)} \mathbb{E}\Big(G\big(u(\omega), \xi(\omega)\big)\Big)$ avec, pour tout u *déterministe*, $G(u, \cdot) = \mathbb{E}\big(J(u, \cdot) \mid \mathcal{H}\big)$.
- Si on approxime $\mathcal{H}$ par la σ-algèbre $\mathcal{H}_k$ engendrée par une *partition finie* $\{\Omega_i\}_{i=1,\dots,k}$, le problème SIS se ramène à un *nombre fini de problèmes en BO*
  - $u(\cdot) \sim \{u_i\}_{i=1,\dots,k}$ *constante par morceaux* sur $\{\Omega_i\}$,
  - $V(\mathcal{P}, \mathcal{H}_k) = \sum_{i=1}^{k} \mathcal{P}(\Omega_i) \times \min_{u_i} \mathbb{E}\Big(J\big(u_i, \xi(\omega)\big) \,\Big|\, \Omega_i\Big)$,

    mais en fait on calculera $V(\mathcal{P}_n, \mathcal{H}_k)$ pour une approximation finie $\mathcal{P}_n$ de $\mathcal{P}$.

guy.cohen@mail.enpc.fr

2.3. Structure d’information dynamique (SID)

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \dots, T$ (horizon fixé) ;

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : "alphabet" ; $\omega_t \in A$;

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : “alphabet” ; $\omega_t \in A$; Ω est l'ensemble des “trajectoires admissibles” ou “mots” produits par le processus stochastique ;

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : “alphabet” ; $\omega_t \in A$; Ω est l'ensemble des “trajectoires admissibles” ou “mots” produits par le processus stochastique ;
- $u = \{u_s\}_{s \leq T-1}$ où u_t désigne la décision prise à t ;

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : “alphabet” ; $\omega_t \in A$; Ω est l'ensemble des “trajectoires admissibles” ou “mots” produits par le processus stochastique ;
- $u = \{u_s\}_{s \leq T-1}$ où u_t désigne la décision prise à t ;
- y_t désigne toutes les observations dont u_t peut dépendre, y compris les observations passées (celles qui sont mémorisées jusqu'à t) ;

Introduction
Typologie des . . .
Un exemple en . . .
Aperçu succinct des . . .
Treillis des partitions . . .
Discrétisation, . . .
Résultats . . .
Conditions . . .

Page 7 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : "alphabet" ; $\omega_t \in A$; Ω est l'ensemble des "trajectoires admissibles" ou "mots" produits par le processus stochastique ;
- $u = \{u_s\}_{s \leq T-1}$ où u_t désigne la décision prise à t ;
- y_t désigne toutes les observations dont u_t peut dépendre, y compris les observations passées (celles qui sont mémorisées jusqu'à t) ;
- la décision u_t dépend de l'observation y_t ;

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : "alphabet" ; $\omega_t \in A$; Ω est l'ensemble des "trajectoires admissibles" ou "mots" produits par le processus stochastique ;
- $u = \{u_s\}_{s \leq T-1}$ où u_t désigne la décision prise à t ;
- y_t désigne toutes les observations dont u_t peut dépendre, y compris les observations passées (celles qui sont mémorisées jusqu'à t) ;
- la décision u_t dépend de l'observation y_t ;
- en général, y_t est une fonction h_t de $\{u_s\}_{s \leq t-1}$ et $\{\omega_s\}_{s \leq t}$:

$$y_t = h_t\big(\omega_t, \omega_{t-1}, \ldots, u_{t-1}, u_{t-2}, \ldots\big) ;$$

guy.cohen@mail.enpc.fr

2.3. Structure d'information dynamique (SID)

Soit $t = 1, \ldots, T$ (horizon fixé) ;

- $\Omega \subset A^T$; A : “alphabet” ; $\omega_t \in A$; Ω est l'ensemble des “trajectoires admissibles” ou “mots” produits par le processus stochastique ;
- $u = \{u_s\}_{s \le T-1}$ où u_t désigne la décision prise à t ;
- y_t désigne toutes les observations dont u_t peut dépendre, y compris les observations passées (celles qui sont mémorisées jusqu'à t) ;
- la décision u_t dépend de l'observation y_t ;
- en général, y_t est une fonction h_t de $\{u_s\}_{s \le t-1}$ et $\{\omega_s\}_{s \le t}$:
$$y_t = h_t\big(\omega_t, \omega_{t-1}, \ldots, u_{t-1}, u_{t-2}, \ldots\big) ;$$
- finalement, u_t doit être mesurable par rapport à la σ-algèbre engendrée par
$$\eta_t^u(\omega) = h_t\big(\omega_t, \omega_{t-1}, \ldots, u_{t-1}(\omega), u_{t-2}(\omega), \ldots\big) .$$

guy.cohen@mail.enpc.fr

2.4. Un exemple

guy.cohen@mail.enpc.fr

2.4. Un exemple

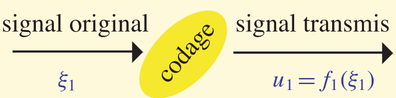
- Un signal aléatoire ξ_1 survient.

$$\xrightarrow[\xi_1]{\text{signal original}}$$

guy.cohen@mail.enpc.fr

2.4. Un exemple

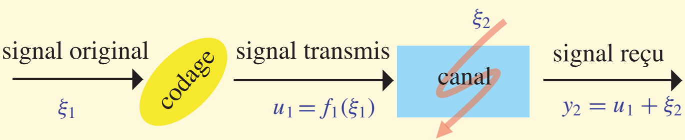
- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal.



guy.cohen@mail.enpc.fr

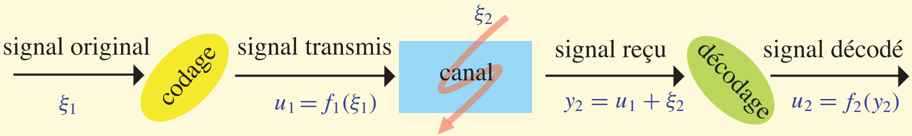
2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2.



2.4. Un exemple

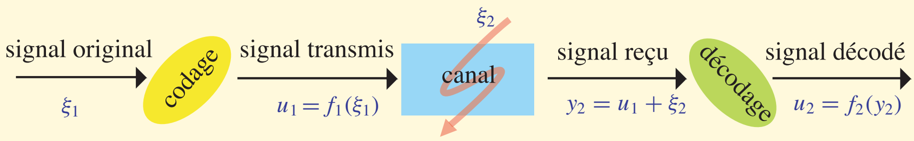
- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).



guy.cohen@mail.enpc.fr

2.4. Un exemple

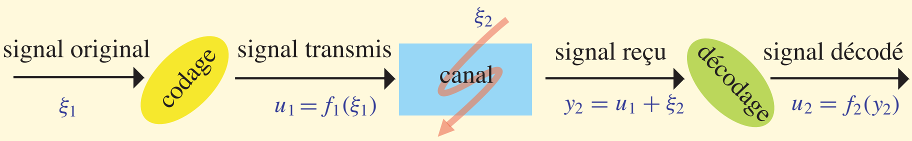
- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$



guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \leq a$

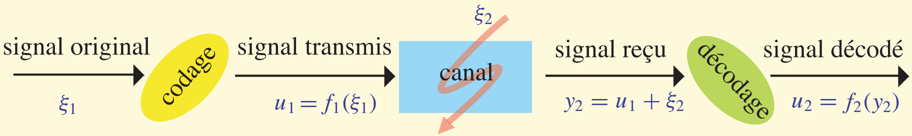


guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \le a$, ou en payant la bande passante :

$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big)$$

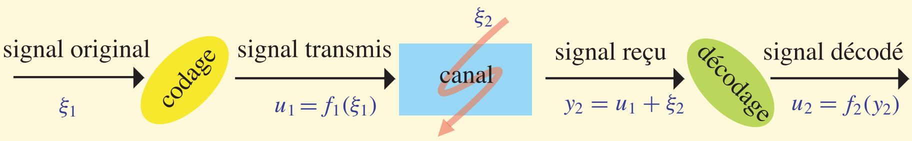


guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \le a$, ou en payant la bande passante :

$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \,.$$



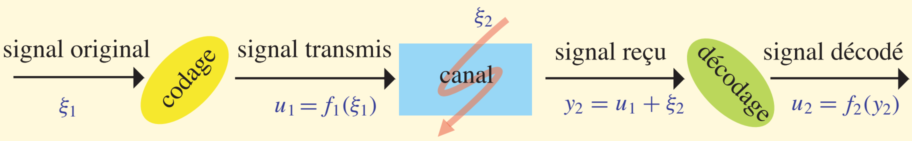
guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \leq a$, ou en payant la bande passante :

$$\min_{u_1, u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \ .$$

- En se limitant à des stratégies de codage linéaires $u_1 = k\xi_1$



guy.cohen@mail.enpc.fr

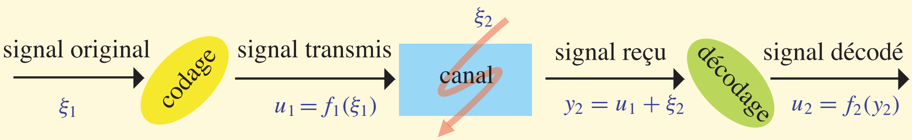
2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).

- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \le a$, ou en payant la bande passante :

$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \ .$$

- En se limitant à des stratégies de codage linéaires $u_1 = k\xi_1$, l'information révélée par y_2 sur ξ_1



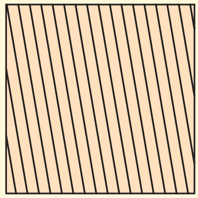
guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \leq a$, ou en payant la bande passante :

$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \; .$$

- En se limitant à des stratégies de codage linéaires $u_1 = k\xi_1$, l'information révélée par y_2 sur ξ_1 est d'autant meilleure que k est grand



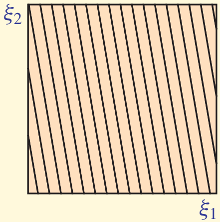
guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).

- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \leq a$, ou en payant la bande passante :
$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \;.$$

- En se limitant à des stratégies de codage linéaires $u_1 = k\xi_1$, l'information révélée par y_2 sur ξ_1 est d'autant meilleure que k est grand, mais cela coûte,



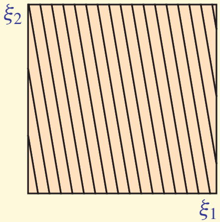
guy.cohen@mail.enpc.fr

2.4. Un exemple

- Un signal aléatoire ξ_1 survient. Le codeur l'observe ($y_1 = \xi_1$) et doit le coder ($u_1 = f_1(\xi_1)$) pour le transmettre sur un canal. Le canal rajoute un bruit ξ_2. Le décodeur observe le signal reçu $y_2 = u_1 + \xi_2$ et doit le décoder ($u_2 = f_2(y_2)$).
- Le but du jeu : trouver u_2 (le signal décodé) le plus proche du signal original ξ_1, donc minimiser $\mathbb{E}(u_2 - \xi_1)^2$ sans dépasser la capacité du canal $\mathbb{E}u_1^2 \leq a$, ou en payant la bande passante :

$$\min_{u_1,u_2} \mathbb{E}\big(\alpha u_1^2 + (u_2 - \xi_1)^2\big) \quad \text{sous} \quad u_1 \preceq \xi_1 \text{ et } u_2 \preceq u_1 + \xi_2 \;.$$

- En se limitant à des stratégies de codage linéaires $u_1 = k\xi_1$, l'information révélée par y_2 sur ξ_1 est d'autant meilleure que k est grand, mais cela coûte, ou viole la contrainte de capacité du canal.



Introduction
Typologie des . . .
Un exemple en . . .
Aperçu succinct des . . .
Treillis des partitions . . .
Discrétisation, . . .
Résultats . . .
Conditions . . .

guy.cohen@mail.enpc.fr

2.5. Absence d'effet dual

guy.cohen@mail.enpc.fr

2.5. Absence d’effet dual

C’est le cas où, pour tout u admissible, la σ-algèbre engendrée par $\eta_t^u(\cdot)$ *ne dépend pas de* u.

guy.cohen@mail.enpc.fr

2.5. Absence d'effet dual

C'est le cas où, pour tout u admissible, la σ-algèbre engendrée par $\eta_t^u(\cdot)$ *ne dépend pas de* u. Dans ce cas, SID se ramène à SIS et, finalement, à une optimisation numérique en BO.

guy.cohen@mail.enpc.fr

2.5. Absence d'effet dual

C'est le cas où, pour tout u admissible, la σ-algèbre engendrée par $\eta_t^u(\cdot)$ *ne dépend pas de* u. Dans ce cas, SID se ramène à SIS et, finalement, à une optimisation numérique en BO.

Sur la caractérisation de l'absence d'effet dual, voir

K. Barty, P. Carpentier, J.-P. Chancelier,
G. Cohen, M. de Lara, T. Guilbaud

Dual effect free stochastic controls

à paraître en 2004 dans

Annals of Operations Research

guy.cohen@mail.enpc.fr

2.5. Absence d'effet dual

C'est le cas où, pour tout u admissible, la σ-algèbre engendrée par $\eta_t^u(\cdot)$ *ne dépend pas de* u. Dans ce cas, SID se ramène à SIS et, finalement, à une optimisation numérique en BO.

Sur la caractérisation de l'absence d'effet dual, voir

K. Barty, P. Carpentier, J.-P. Chancelier,
G. Cohen, M. de Lara, T. Guilbaud

Dual effect free stochastic controls

à paraître en 2004 dans

Annals of Operations Research

aussi disponible sur
STOCHASTIC PROGRAMMING E-PRINT SERIES
Année 2003, papier No. 18
`http://www.speps.info`

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ;

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ;

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ;

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ;

Page 10 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x.

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

On suppose que $\varepsilon > 0$ est "petit".

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

3.2. Résolution exacte

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

3.2. Résolution exacte

Ce problème *LQG — Linéaire-Quadratique-Gaussien —* avec bruits centrés admet une décision optimale de la forme $u = cx$;

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

3.2. Résolution exacte

Ce problème *LQG — Linéaire-Quadratique-Gaussien —* avec bruits centrés admet une décision optimale de la forme $u = cx$; le feedback et le coût optimaux sont

$$\text{feedback} \quad c = -\frac{1}{\varepsilon + 1} \; ; \qquad \text{coût} \quad \frac{1}{\varepsilon + 1}\left(\varepsilon\sigma_x^2 + (\varepsilon + 1)\sigma_w^2\right) ,$$

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

3.2. Résolution exacte

Ce problème *LQG — Linéaire-Quadratique-Gaussien —* avec bruits centrés admet une décision optimale de la forme $u = cx$; le feedback et le coût optimaux sont

$$\text{feedback} \quad c = -\frac{1}{\varepsilon + 1} \; ; \qquad \text{coût} \quad \frac{1}{\varepsilon + 1}\left(\varepsilon\sigma_x^2 + (\varepsilon + 1)\sigma_w^2\right) ,$$

donc le coût est de l'ordre de σ_w^2 (disons d'ordre 1) ;

guy.cohen@mail.enpc.fr

3. Un exemple en commande stochastique

3.1. Le problème

- x est une condition initiale aléatoire (loi normale, moyenne nulle, écart-type σ_x) ; u est une variable de décision basée sur l'observation x ; w est une autre variable aléatoire (loi normale, moyenne nulle, écart-type σ_w) ; x et w sont indépendants ; $z = x + u + w$ est l'"état final".
- Le but est de minimiser la fonction coût $\mathbb{E}(\varepsilon u^2 + z^2)$ en choisissant u comme fonction de x. Autrement dit, $\xi = (x, w)$ et $y = x$.

3.2. Résolution exacte

Ce problème *LQG — Linéaire-Quadratique-Gaussien —* avec bruits centrés admet une décision optimale de la forme $u = cx$; le feedback et le coût optimaux sont

$$\text{feedback} \quad c = -\frac{1}{\varepsilon + 1} \; ; \qquad \text{coût} \quad \frac{1}{\varepsilon + 1}\big(\varepsilon\sigma_x^2 + (\varepsilon + 1)\sigma_w^2\big) \; ,$$

donc le coût est de l'ordre de σ_w^2 (disons d'ordre 1) ; intuitivement, puisque w agit *après* la décision u, le terme $\mathbb{E}z^2$ est de l'ordre de σ_w^2.

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \dots, N$.

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \ldots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \; .$$

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \dots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \; .$$

La minimisation de cette expression en u donne :

$$\text{décision } \; u_i = -\frac{x_i + w_i}{\varepsilon + 1} \; ; \qquad \text{coût } \; \frac{\varepsilon (x_i + w_i)^2}{\varepsilon + 1} \; .$$

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \ldots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \; .$$

La minimisation de cette expression en u donne :

$$\text{décision} \;\; u_i = -\frac{x_i + w_i}{\varepsilon + 1} \; ; \qquad \text{coût} \;\; \frac{\varepsilon (x_i + w_i)^2}{\varepsilon + 1} \; .$$

Le coût est d'ordre ε

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \dots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \ .$$

La minimisation de cette expression en u donne :

$$\text{décision} \ \ u_i = -\frac{x_i + w_i}{\varepsilon + 1} \ ; \qquad \text{coût} \ \ \frac{\varepsilon (x_i + w_i)^2}{\varepsilon + 1} \ .$$

Le coût est d'ordre ε et ceci reste vrai pour la moyenne sur N réalisations, aussi grand que soit N !

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \ldots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \; .$$

La minimisation de cette expression en u donne :

$$\text{décision } \; u_i = -\frac{x_i + w_i}{\varepsilon + 1} \; ; \qquad \text{coût } \; \frac{\varepsilon (x_i + w_i)^2}{\varepsilon + 1} \; .$$

Le coût est d'ordre ε et ceci reste vrai pour la moyenne sur N réalisations, aussi grand que soit N !

Explication : chaque réalisation utilise sa propre décision u_i qui dépend non seulement de x_i mais aussi de w_i, et c'est donc une décision *anticipative*.

guy.cohen@mail.enpc.fr

3.3. Méthode de Monte Carlo naïve

On se donne N réalisations (x_i, w_i), $i = 1, \ldots, N$. Pour chaque réalisation, le coût vaut

$$\varepsilon u^2 + (x_i + u + w_i)^2 = (\varepsilon + 1)u^2 + 2(x_i + w_i)u + (x_i + w_i)^2 \ .$$

La minimisation de cette expression en u donne :

$$\text{décision} \ \ u_i = -\frac{x_i + w_i}{\varepsilon + 1} \ ; \qquad \text{coût} \ \ \frac{\varepsilon (x_i + w_i)^2}{\varepsilon + 1} \ .$$

Le coût est d'ordre ε et ceci reste vrai pour la moyenne sur N réalisations, aussi grand que soit N !

Explication : chaque réalisation utilise sa propre décision u_i qui dépend non seulement de x_i mais aussi de w_i, et c'est donc une décision *anticipative*.

Pratiquement, **il est inutile** d'imposer la contrainte que

$$\forall (i, j) : x_i = x_j \Rightarrow u_i = u_j \ . \tag{1}$$

4. Aperçu succinct des résultats de convergence

guy.cohen@mail.enpc.fr

4. Aperçu succinct des résultats de convergence

4.1. Cadre

guy.cohen@mail.enpc.fr

4. Aperçu succinct des résultats de convergence

4.1. Cadre

- On cherche à résoudre

$$V(\mathcal{P}, \mathcal{H}) = \min_{u} \mathbb{E}_{\mathcal{P}} J\big(u(\omega), \xi(\omega)\big) \text{ sous } u \ \mathcal{H}\text{-mesurable.}$$

guy.cohen@mail.enpc.fr

4. Aperçu succinct des résultats de convergence

4.1. Cadre

- On cherche à résoudre

$$V(\mathcal{P},\mathcal{H}) = \min_u \mathbb{E}_{\mathcal{P}} J\big(u(\omega),\xi(\omega)\big) \text{ sous } u \ \mathcal{H}\text{-mesurable.}$$

- On résout à la place

$$V(\mathcal{P}_n,\mathcal{H}_k) = \min_u \mathbb{E}_{\mathcal{P}_n} J\big(u(\omega),\xi(\omega)\big) \text{ sous } u \ \mathcal{H}_k\text{-mesurable.}$$

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,

$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\ ;$$

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,

   $$\lim_{k \to +\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0 \;;$$

   elle est obtenue avec la métrique de Cotter ;

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\le 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$

Page 13 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\le 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec les métriques de Boylan, Rogge, Barty.

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\le 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec les métriques de Boylan, Rogge, Barty.

Topologies équivalentes pour des espaces de probabilité purement atomiques

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\leq 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec les métriques de Boylan, Rogge, Barty.

Topologies équivalentes pour des espaces de probabilité purement atomiques

- Les σ-algèbres *engendrées par des partitions finies* de Ω sont *denses* avec 1 mais pas avec 2.

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0 ;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\le 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0 ;$$
elle est obtenue avec les métriques de Boylan, Rogge, Barty.

Topologies équivalentes pour des espaces de probabilité purement atomiques

- Les σ-algèbres *engendrées par des partitions finies* de Ω sont *denses* avec 1 mais pas avec 2.
- Si une suite de v.a. $\{X_n\}$ tend vers X *en probabilité* et si $\sigma(X_n) \subset \sigma(X)$, alors $\sigma(X_n)$ tend vers $\sigma(X)$ au sens de 1 (continuité *par dessous*), pas au sens de 2.

guy.cohen@mail.enpc.fr

4.2. Topologies sur les tribus

Deux topologies pour la convergence de $\mathcal{H}_k$ vers $\mathcal{H}$:

1. l'une assure que $\forall f \in L^1(\Omega)$, $|f|$ bornée,
$$\lim_{k\to+\infty} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec la métrique de Cotter ;

2. l'autre assure que
$$\lim_{k\to+\infty} \sup_{\|f\|\le 1} \|\mathbb{E}(f \mid \mathcal{H}_k) - \mathbb{E}(f \mid \mathcal{H})\|_{L_1} = 0\,;$$
elle est obtenue avec les métriques de Boylan, Rogge, Barty.

Topologies équivalentes pour des espaces de probabilité purement atomiques

- Les σ-algèbres *engendrées par des partitions finies* de Ω sont *denses* avec 1 mais pas avec 2.
- Si une suite de v.a. $\{X_n\}$ tend vers X *en probabilité* et si $\sigma(X_n) \subset \sigma(X)$, alors $\sigma(X_n)$ tend vers $\sigma(X)$ au sens de 1 (continuité *par dessous*), pas au sens de 2.
- L'opération $\vee$ entre tribus (tribu engendrée par l'union de tribus) est continue *par dessous* avec 1 (toujours avec 2).

guy.cohen@mail.enpc.fr

4.3. Résultats

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

  Par exemple, si $\mathcal{H}$ est engendrée par une observation $y : \Omega \to \mathcal{Y}$, $\mathcal{H}_k$ peut être engendrée par $Q_k^{\mathcal{Y}} \circ y$ où $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ a une *image finie*.

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

  Par exemple, si $\mathcal{H}$ est engendrée par une observation $y : \Omega \to \mathcal{Y}$, $\mathcal{H}_k$ peut être engendrée par $Q_k^{\mathcal{Y}} \circ y$ où $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ a une *image finie*. On suppose que $Q_k^{\mathcal{Y}}$ tend vers l'identité de sorte que $Q_k^{\mathcal{Y}} \circ y$ tend vers y en probabilité.

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

Alors

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

Alors

- D'une part, si $\mathcal{P}$ ne charge pas les frontières de la partition $\mathcal{H}_k$

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

Alors

- D'une part, si $\mathcal{P}$ ne charge pas les frontières de la partition $\mathcal{H}_k$, alors $\lim_{n\to+\infty} V(\mathcal{P}_n, \mathcal{H}_k) = V(\mathcal{P}, \mathcal{H}_k)$ pour tout k.

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

Alors

- D'une part, si $\mathcal{P}$ ne charge pas les frontières de la partition $\mathcal{H}_k$, alors $\lim_{n\to+\infty} V(\mathcal{P}_n, \mathcal{H}_k) = V(\mathcal{P}, \mathcal{H}_k)$ pour tout k.
- D'autre part, $\lim_{k\to+\infty} V(\mathcal{P}, \mathcal{H}_k) = V(\mathcal{P}, \mathcal{H})$.

guy.cohen@mail.enpc.fr

4.3. Résultats

Hypothèses

- Hypothèses techniques sur J.
- On construit une suite $\mathcal{P}_n$ de lois de probabilité *convergeant en loi* vers $\mathcal{P}$ (par exemple des lois atomiques basées sur des tirages aléatoires — Monte Carlo).
- On construit une suite *croissante* de σ-algèbres $\mathcal{H}_k \subset \mathcal{H}$ engendrées par des partitions finies $\{\Omega_i\}_k$ de Ω. On suppose que $\mathcal{H}_k$ tend vers $\mathcal{H}$ au sens de Cotter (1).

Alors

- D'une part, si $\mathcal{P}$ ne charge pas les frontières de la partition $\mathcal{H}_k$, alors $\lim_{n \to +\infty} V(\mathcal{P}_n, \mathcal{H}_k) = V(\mathcal{P}, \mathcal{H}_k)$ pour tout k.
- D'autre part, $\lim_{k \to +\infty} V(\mathcal{P}, \mathcal{H}_k) = V(\mathcal{P}, \mathcal{H})$.
- Corollaire : il existe une suite croissante $\{k(n)\}$ telle que

$$\lim_{n \to +\infty} V(\mathcal{P}_n, \mathcal{H}_{k(n)}) = V(\mathcal{P}, \mathcal{H}) .$$

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$.

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *“mesurable”* par rapport à g si l’une des conditions *équivalentes* suivantes est vérifiée :

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *"mesurable"* par rapport à g si l'une des conditions *équivalentes* suivantes est vérifiée :

- $\forall \omega \in \Omega, \forall \omega' \in \Omega, \quad g(\omega) = g(\omega') \implies f(\omega) = f(\omega')$, c'est-à-dire que l'observation g est plus forte (pas plus faible) que l'observation f si, lorsque ω et ω' ne peuvent être distingués par g, ils ne peuvent pas non plus être distingués par f ;

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *"mesurable"* par rapport à g si l'une des conditions *équivalentes* suivantes est vérifiée :

- $\forall \omega \in \Omega, \forall \omega' \in \Omega, \quad g(\omega) = g(\omega') \implies f(\omega) = f(\omega')$, c'est-à-dire que l'observation g est plus forte (pas plus faible) que l'observation f si, lorsque ω et ω' ne peuvent être distingués par g, ils ne peuvent pas non plus être distingués par f ;
- $\exists p : \operatorname{im} g \to \operatorname{im} f : f = p \circ g$, c'est-à-dire que f est une fonction de g.

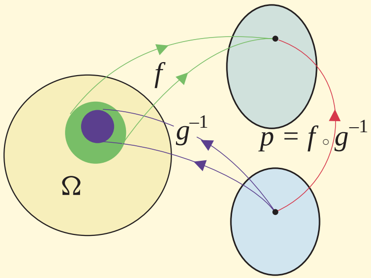
guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *“mesurable”* par rapport à g si l'une des conditions *équivalentes* suivantes est vérifiée :

- $\forall \omega \in \Omega, \forall \omega' \in \Omega, \quad g(\omega) = g(\omega') \implies f(\omega) = f(\omega')$, c'est-à-dire que l'observation g est plus forte (pas plus faible) que l'observation f si, lorsque ω et ω' ne peuvent être distingués par g, ils ne peuvent pas non plus être distingués par f ;
- $\exists p : \operatorname{im} g \to \operatorname{im} f : f = p \circ g$, c'est-à-dire que f est une fonction de g.



guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *"mesurable"* par rapport à g si l'une des conditions *équivalentes* suivantes est vérifiée :

- $\forall \omega \in \Omega, \forall \omega' \in \Omega, \quad g(\omega) = g(\omega') \implies f(\omega) = f(\omega')$, c'est-à-dire que l'observation g est plus forte (pas plus faible) que l'observation f si, lorsque ω et ω' ne peuvent être distingués par g, ils ne peuvent pas non plus être distingués par f ;
- $\exists p : \operatorname{im} g \to \operatorname{im} f : f = p \circ g$, c'est-à-dire que f est une fonction de g.

Cette relation est un *pré-ordre* sur les fonctions sur Ω ;

guy.cohen@mail.enpc.fr

5. Treillis des partitions et quantification

5.1. Une équivalence de fonctions sur Ω et un pré-ordre

Soit $f : \Omega \to \mathcal{F}$ et $g : \Omega \to \mathcal{G}$. Alors, $g \succeq f$ ou $f \preceq g$ ou f est *"mesurable"* par rapport à g si l'une des conditions *équivalentes* suivantes est vérifiée :

- $\forall \omega \in \Omega, \forall \omega' \in \Omega, \quad g(\omega) = g(\omega') \implies f(\omega) = f(\omega')$, c'est-à-dire que l'observation g est plus forte (pas plus faible) que l'observation f si, lorsque ω et ω' ne peuvent être distingués par g, ils ne peuvent pas non plus être distingués par f ;
- $\exists p : \operatorname{im} g \to \operatorname{im} f : f = p \circ g$, c'est-à-dire que f est une fonction de g.

Cette relation est un *pré-ordre* sur les fonctions sur Ω ; c'est un ordre pour les *classes d'équivalence de fonctions* sur Ω : f et g sont *équivalentes* ($f \sim g$) si

$$\{f \preceq g \textbf{ et } g \preceq f\} \iff \{g(\omega) = g(\omega') \Leftrightarrow f(\omega) = f(\omega')\}$$
$$\iff \exists p \textbf{ injective (bijective)} : \operatorname{im} g \to \operatorname{im} f : f = p \circ g .$$

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h.$

guy.cohen@mail.enpc.fr

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h.$

5.2. Partitions

guy.cohen@mail.enpc.fr

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h$.

5.2. Partitions

- $f : \Omega \to \mathcal{F}$ définit une *relation d'équivalence sur* Ω :

$$\omega \overset{f}{\equiv} \omega' \iff f(\omega) = f(\omega') \iff \omega \in f^{-1}\big(f(\omega')\big) .$$

guy.cohen@mail.enpc.fr

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h$.

5.2. Partitions

- $f : \Omega \to \mathcal{F}$ définit une *relation d'équivalence sur* Ω :

$$\omega \overset{f}{\equiv} \omega' \iff f(\omega) = f(\omega') \iff \omega \in f^{-1}\bigl(f(\omega')\bigr) .$$

- La *partition* de Ω en (f-)classes d'équivalence est notée Ω/f ; cette partition est en bijection avec $\operatorname{im} f$.

guy.cohen@mail.enpc.fr

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h.$

5.2. Partitions

- $f : \Omega \to \mathcal{F}$ définit une *relation d'équivalence sur* Ω :

$$\omega \overset{f}{\equiv} \omega' \iff f(\omega) = f(\omega') \iff \omega \in f^{-1}\big(f(\omega')\big) .$$

- La *partition* de Ω en (f-)classes d'équivalence est notée Ω/f ; cette partition est en bijection avec $\operatorname{im} f$.

- $f \preceq g$ $\iff$ toute f-classe est l'union de g-classes ;
  $\iff$ toute g-classe est incluse dans une f-classe ;
  $\iff$ Ω/g est *plus fine* que Ω/f.

guy.cohen@mail.enpc.fr

Note : $f \preceq g \Longrightarrow \forall h : \Omega' \to \Omega, \quad f \circ h \preceq g \circ h.$

5.2. Partitions

- $f : \Omega \to \mathcal{F}$ définit une *relation d'équivalence sur* Ω :

$$\omega \overset{f}{\equiv} \omega' \iff f(\omega) = f(\omega') \iff \omega \in f^{-1}\big(f(\omega')\big) .$$

- La *partition* de Ω en (f-)classes d'équivalence est notée Ω/f ; cette partition est en bijection avec $\mathrm{im}\ f$.

- $f \preceq g \iff$ toute f-classe est l'union de g-classes ;
  $\iff$ toute g-classe est incluse dans une f-classe ;
  $\iff$ Ω/g est *plus fine* que Ω/f.

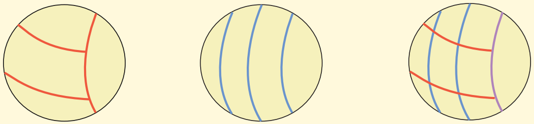
- **Un représentant spécial :** pour tout $f : \Omega \to \mathcal{F}$, il existe $\Pi : \Omega \to \Omega$ tel que $\Pi \circ \Pi = \Pi$ (*projection*) et $\Pi \sim f$; essentiellement, Π applique tout ω sur un $\overline{\omega}$ dans la même f-classe que ω.

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.

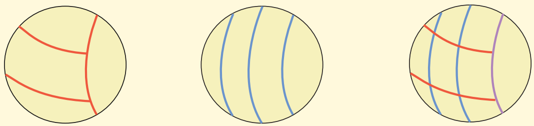


f-classes g-classes $(f \vee g)$-classes

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes — g-classes — $(f \vee g)$-classes

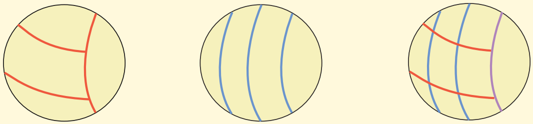
Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

Introduction
Typologie des . . .
Un exemple en . . .
Aperçu succinct des . . .
Treillis des partitions . . .
Discrétisation, . . .
Résultats . . .
Conditions . . .
Page 17 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes $\quad$ g-classes $\quad$ $(f \vee g)$-classes

Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

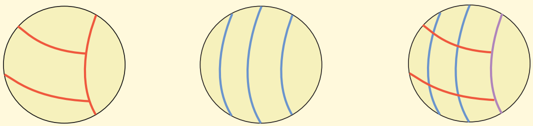
Top (noté $\top$) : la classe des fonctions *injectives* sur Ω.

Plein écran
Zoom Main
Copier Coller
Signets Dest.

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes $\quad$ g-classes $\quad$ $(f \vee g)$-classes

Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

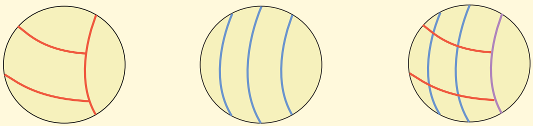
Top (noté $\top$) : la classe des fonctions *injectives* sur Ω.

Continuité ? Soit $X : \Omega \to \{1, \ldots, p\}$ et soit $\{X_n = X/n\}_{n \in \mathbb{N}}$.

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes $\quad$ g-classes $\quad$ $(f \vee g)$-classes

Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

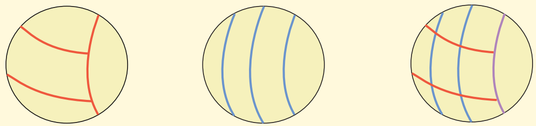
Top (noté $\top$) : la classe des fonctions *injectives* sur Ω.

Continuité ? Soit $X : \Omega \to \{1, \dots, p\}$ et soit $\{X_n = X/n\}_{n\in\mathbb{N}}$. Pour tout n, $\Omega/X_n \sim \Omega/X$

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes $\qquad$ g-classes $\qquad$ $(f \vee g)$-classes

Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

Top (noté $\top$) : la classe des fonctions *injectives* sur Ω.

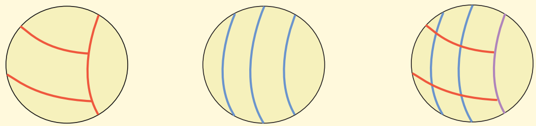
Continuité ? Soit $X : \Omega \to \{1, \dots, p\}$ et soit $\{X_n = X/n\}_{n \in \mathbb{N}}$. Pour tout n, $\Omega/X_n \sim \Omega/X$ mais $\Omega/(\lim_{n \to +\infty} X_n) =$ la partition grossière.

Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

5.3. Opérations sur le treillis des partitions

Plus petite borne sup : la plus grossière des partitions plus fine que Ω/f et Ω/g ; elle est engendrée par $h : \Omega \to \mathcal{F} \times \mathcal{G}$ telle que $h(\omega) = \big(f(\omega), g(\omega)\big)$; h est un représentant de $f \vee g$.



f-classes $\quad$ g-classes $\quad$ $(f \vee g)$-classes

Bottom (noté $\perp$) : la classe des fonctions *constantes* sur Ω.

Top (noté $\top$) : la classe des fonctions *injectives* sur Ω.

Continuité ? Soit $X : \Omega \to \{1, \ldots, p\}$ et soit $\{X_n = X/n\}_{n \in \mathbb{N}}$. Pour tout n, $\Omega/X_n \sim \Omega/X$ mais $\Omega/(\lim_{n\to+\infty} X_n) =$ la partition grossière. Noter que $X_n \not\preceq \lim_{n\to+\infty} X_n$.

guy.cohen@mail.enpc.fr

5.4. Quantification

guy.cohen@mail.enpc.fr

5.4. Quantification

Notions ensemblistes

guy.cohen@mail.enpc.fr

5.4. Quantification

Notions ensemblistes

- q est une projection de Ω sur lui-même à image *de cardinal fini* N (cette image est appelée le *codebook*) ;

guy.cohen@mail.enpc.fr

5.4. Quantification

Notions ensemblistes

- q est une projection de Ω sur lui-même à image *de cardinal fini* N (cette image est appelée le *codebook*) ;
- q est factorisée en $d \circ e$ avec
  - $e : \Omega \to \{1, \ldots, N\}$ appelée *codage* ;
  - $d : \{1, \ldots, N\} \to \operatorname{im} q \subset \Omega$, *bijective*, appelée *décodage* ;

guy.cohen@mail.enpc.fr

5.4. Quantification

Notions ensemblistes

- q est une projection de Ω sur lui-même à image *de cardinal fini* N (cette image est appelée le *codebook*) ;
- q est factorisée en $d \circ e$ avec
  - $e : \Omega \to \{1, \ldots, N\}$ appelée *codage* ;
  - $d : \{1, \ldots, N\} \to \operatorname{im} q \subset \Omega$, *bijective*, appelée *décodage* ;
- seule e importe pour caractériser q du point de vue de la mesurabilité ($q \sim e$) ; $\Omega/q \approx \Omega/e$ définit les *cellules* de la quantification ;

guy.cohen@mail.enpc.fr

5.4. Quantification

Notions ensemblistes

- q est une projection de Ω sur lui-même à image *de cardinal fini* N (cette image est appelée le *codebook*) ;
- q est factorisée en $d \circ e$ avec
  - $e : \Omega \to \{1, \dots, N\}$ appelée *codage* ;
  - $d : \{1, \dots, N\} \to \operatorname{im} q \subset \Omega$, *bijective*, appelée *décodage* ;
- seule e importe pour caractériser q du point de vue de la mesurabilité ($q \sim e$) ; $\Omega/q \approx \Omega/e$ définit les *cellules* de la quantification ;
- d définit un *représentant* (élément de Ω) de chaque cellule (appartenant lui-même à cette cellule).

guy.cohen@mail.enpc.fr

Optimalité

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$,

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_\Omega \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\big(\|\overline{\omega}_i - \omega\|^2 \,\big|\, \Omega_i\big) \times \mathcal{P}(\Omega_i) \; .$$

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_{\Omega} \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\Big(\|\overline{\omega}_i - \omega\|^2 \,\Big|\, \Omega_i\Big) \times \mathcal{P}(\Omega_i) \ .$$

- Deux conditions nécessaires d'optimalité :

guy.cohen@mail.enpc.fr

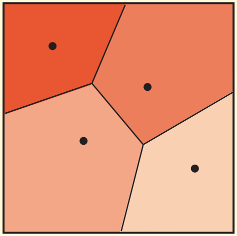
Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_\Omega \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\Big(\|\overline{\omega}_i - \omega\|^2 \;\Big|\; \Omega_i\Big) \times \mathcal{P}(\Omega_i) \; .$$

- Deux conditions nécessaires d'optimalité :

  1. pour tout ω, la distance de $q(\omega)$ est inférieure ou égale à la distance à tout autre élément du codebook $q(\omega')$ ("cellules de Voronoi") ;



guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'“optimalité” de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_{\Omega} \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\Big(\|\overline{\omega}_i - \omega\|^2 \;\Big|\; \Omega_i\Big) \times \mathcal{P}(\Omega_i) \; .$$

- Deux conditions nécessaires d'optimalité :

  1. pour tout ω, la distance de $q(\omega)$ est inférieure ou égale à la distance à tout autre élément du codebook $q(\omega')$ (“cellules de Voronoi”) ;

  2. tout élément du codebook $\overline{\omega}_i$ est l'espérance conditionnelle de ω sachant que ω appartient à la cellule Ω_i.

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_{\Omega} \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\Big(\|\overline{\omega}_i - \omega\|^2 \,\Big|\, \Omega_i\Big) \times \mathcal{P}(\Omega_i) \; .$$

- Deux conditions nécessaires d'optimalité :

  1. pour tout ω, la distance de $q(\omega)$ est inférieure ou égale à la distance à tout autre élément du codebook $q(\omega')$ ("cellules de Voronoi") ;

  2. tout élément du codebook $\overline{\omega}_i$ est l'espérance conditionnelle de ω sachant que ω appartient à la cellule Ω_i.

- Erreur quadratique moyenne minimale $+$ variance de la variable "quantifiée" $=$ variance de ω

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_{\Omega} \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\Big(\|\overline{\omega}_i - \omega\|^2 \;\Big|\; \Omega_i \Big) \times \mathcal{P}(\Omega_i) \; .$$

- Deux conditions nécessaires d'optimalité :

  1. pour tout ω, la distance de $q(\omega)$ est inférieure ou égale à la distance à tout autre élément du codebook $q(\omega')$ ("cellules de Voronoi") ;

  2. tout élément du codebook $\overline{\omega}_i$ est l'espérance conditionnelle de ω sachant que ω appartient à la cellule Ω_i.

- Erreur quadratique moyenne minimale $+$ variance de la variable "quantifiée" $=$ variance de ω (donc le moment d'ordre 2 de la variable quantifiée est *sous-estimé* par rapport à sa vraie valeur ;

Page 19 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

guy.cohen@mail.enpc.fr

Optimalité

- Si Ω est un espace vectoriel normé, avec éventuellement une mesure de probabilité $\mathcal{P}$, l'"optimalité" de la quantification est définie par l'idée d'*erreur quadratique moyenne* :

$$\int_\Omega \|q(\omega) - \omega\|^2 \, \mathcal{P}(d\omega) = \sum_{i=1}^{N} \mathbb{E}\big(\|\overline{\omega}_i - \omega\|^2 \;\big|\; \Omega_i\big) \times \mathcal{P}(\Omega_i) \; .$$

- Deux conditions nécessaires d'optimalité :

  1. pour tout ω, la distance de $q(\omega)$ est inférieure ou égale à la distance à tout autre élément du codebook $q(\omega')$ ("cellules de Voronoi") ;

  2. tout élément du codebook $\overline{\omega}_i$ est l'espérance conditionnelle de ω sachant que ω appartient à la cellule Ω_i.

- Erreur quadratique moyenne minimale $+$ variance de la variable "quantifiée" $=$ variance de ω (donc le moment d'ordre 2 de la variable quantifiée est *sous-estimé* par rapport à sa vraie valeur ; le moment d'ordre 1 — l'espérance — est correctement représenté).

6. Discrétisation, modalités pratiques

guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.

guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.
- On rappelle que $y = x$ (x sera représenté en abscisse, w en ordonnée).

guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.
- On rappelle que $y = x$ (x sera représenté en abscisse, w en ordonnée).

6.1. Discrétisation de ξ

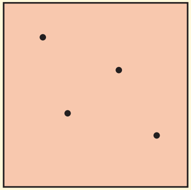
guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.
- On rappelle que $y = x$ (x sera représenté en abscisse, w en ordonnée).

6.1. Discrétisation de ξ

- Échantillonnage à la Monte Carlo.



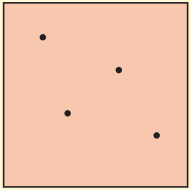
guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.
- On rappelle que $y = x$ (x sera représenté en abscisse, w en ordonnée).

6.1. Discrétisation de ξ

- Échantillonnage à la Monte Carlo.
- Remplacer $\xi(\cdot) : \Omega \to \Xi$ par $\widetilde{\xi}(\cdot) = Q^{\Xi}\big(\xi(\cdot)\big)$ où $Q^{\Xi} : \Xi \to \Xi$ est une quantification de Ξ.



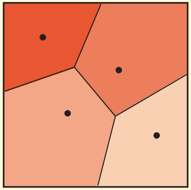
guy.cohen@mail.enpc.fr

6. Discrétisation, modalités pratiques

- On reprend l'exemple §3 et on suppose que $\xi = (x, w)$ appartient à $[-1, 1] \times [-1, 1]$.
- On rappelle que $y = x$ (x sera représenté en abscisse, w en ordonnée).

6.1. Discrétisation de ξ

- Échantillonnage à la Monte Carlo.
- Remplacer $\xi(\cdot) : \Omega \to \Xi$ par $\widetilde{\xi}(\cdot) = Q^{\Xi}\big(\xi(\cdot)\big)$ où $Q^{\Xi} : \Xi \to \Xi$ est une quantification de Ξ.
- Combiner ces deux approches : les échantillons servent de "codebook" et on complète la quantification par la constitution de cellules (de Voronoi) autour des échantillons.



guy.cohen@mail.enpc.fr

6.2. Discrétisation de y

6.2. Discrétisation de y

Quantification de y en liaison avec celle de ξ

guy.cohen@mail.enpc.fr

6.2. Discrétisation de y

Quantification de y en liaison avec celle de ξ

- Dans l'exemple §3, on avait défini $\widetilde{y} = h(\widetilde{\xi}) = \widetilde{x}$ et on avait envisagé d'imposer la contrainte $u \preceq \widetilde{y}$ (voir (1)).

guy.cohen@mail.enpc.fr

6.2. Discrétisation de y

Quantification de y en liaison avec celle de ξ

- Dans l'exemple §3, on avait défini $\widetilde{y} = h(\widetilde{\xi}) = \widetilde{x}$ et on avait envisagé d'imposer la contrainte $u \preceq \widetilde{y}$ (voir (1)).
- Mais si $\widetilde{\xi} \preceq \xi$, il *ne s'ensuit pas* que $h \circ \widetilde{\xi} \preceq h \circ \xi$.

guy.cohen@mail.enpc.fr

6.2. Discrétisation de y

Quantification de y en liaison avec celle de ξ

- Dans l'exemple §3, on avait défini $\widetilde{y} = h(\widetilde{\xi}) = \widetilde{x}$ et on avait envisagé d'imposer la contrainte $u \preceq \widetilde{y}$ (voir (1)).
- Mais si $\widetilde{\xi} \preceq \xi$, il *ne s'ensuit pas* que $h \circ \widetilde{\xi} \preceq h \circ \xi$.
- Donc $u \preceq \widetilde{y}$ n'implique pas $u \preceq y$, c'est-à-dire que l'approximation de y par $\widetilde{y}$ ne se fait pas par dessous !

guy.cohen@mail.enpc.fr

6.2. Discrétisation de y

Quantification de y en liaison avec celle de ξ

- Dans l'exemple §3, on avait défini $\widetilde{y} = h(\widetilde{\xi}) = \widetilde{x}$ et on avait envisagé d'imposer la contrainte $u \preceq \widetilde{y}$ (voir (1)).
- Mais si $\widetilde{\xi} \preceq \xi$, il *ne s'ensuit pas* que $h \circ \widetilde{\xi} \preceq h \circ \xi$.
- Donc $u \preceq \widetilde{y}$ n'implique pas $u \preceq y$, c'est-à-dire que l'approximation de y par $\widetilde{y}$ ne se fait pas par dessous !
- Dans l'exemple on avait même (avec probabilité 1) $\widetilde{y} \sim \widetilde{\xi}$: le "bruit" était entièrement prévisible à partir de l'observation (anticipativité).

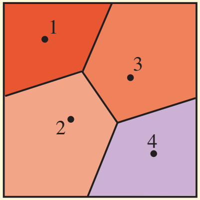
guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

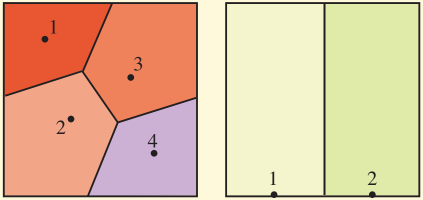
- Quantification $Q^{\Xi} : \Xi \to \Xi$ et $\widetilde{\xi} = Q^{\Xi} \circ \xi$.



guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

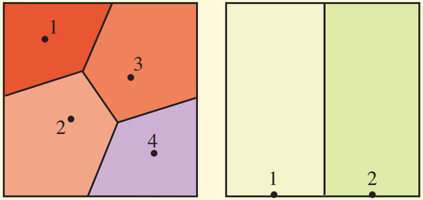
- Quantification $Q^{\Xi} : \Xi \to \Xi$ et $\widetilde{\xi} = Q^{\Xi} \circ \xi$.
- Quantification $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ et $\widetilde{y} = Q^{\mathcal{Y}} \circ y$.



guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

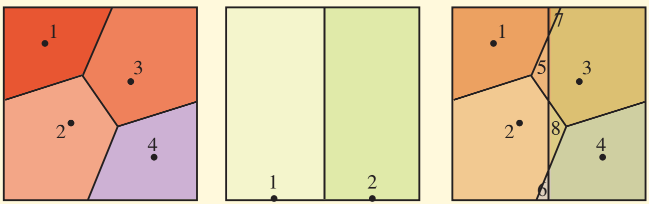
- Quantification $Q^{\Xi} : \Xi \to \Xi$ et $\widetilde{\xi} = Q^{\Xi} \circ \xi$.
- Quantification $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ et $\widetilde{y} = Q^{\mathcal{Y}} \circ y$.
- On résout donc le problème numérique *fini* $\min_{u \preceq \widetilde{y}} \mathbb{E} J\big(u(\omega), \widetilde{\xi}(\omega)\big)$.



guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

- Quantification $Q^{\Xi} : \Xi \to \Xi$ et $\widetilde{\xi} = Q^{\Xi} \circ \xi$.
- Quantification $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ et $\widetilde{y} = Q^{\mathcal{Y}} \circ y$.
- On résout donc le problème numérique *fini* $\min\limits_{u \preceq \widetilde{y}} \mathbb{E} J\big(u(\omega), \widetilde{\xi}(\omega)\big)$.
- Le coût est mesurable sur la σ-algèbre engendrée par la *partition finie* $\Omega/(\widetilde{\xi} \vee \widetilde{y})$.

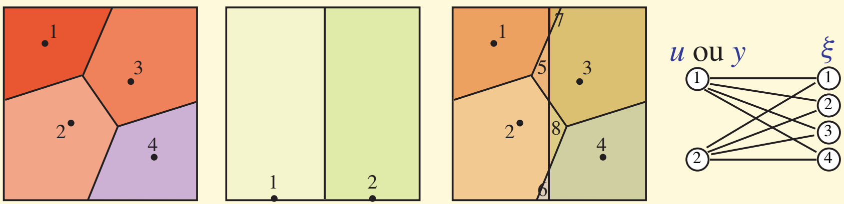


guy.cohen@mail.enpc.fr

Quantifications indépendantes de ξ et de y

- Quantification $Q^{\Xi} : \Xi \to \Xi$ et $\widetilde{\xi} = Q^{\Xi} \circ \xi$.
- Quantification $Q^{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ et $\widetilde{y} = Q^{\mathcal{Y}} \circ y$.
- On résout donc le problème numérique *fini* $\min\limits_{u \preceq \widetilde{y}} \mathbb{E} J\big(u(\omega), \widetilde{\xi}(\omega)\big)$.
- Le coût est mesurable sur la σ-algèbre engendrée par la *partition finie* $\Omega/(\widetilde{\xi} \vee \widetilde{y})$.
- On doit résoudre

$$\min_{u_1,u_2} \Bigg(\sum_{j\in\{1,2,5,6\}} \mathcal{P}(\Omega_j) \times J(u_1, \widetilde{\xi}_{1\text{ à }4}) + \sum_{k\in\{3,4,7,8\}} \mathcal{P}(\Omega_k) \times J(u_2, \widetilde{\xi}_{1\text{ à }4}) \Bigg) .$$



Récupération d'une fonction d'observation

guy.cohen@mail.enpc.fr

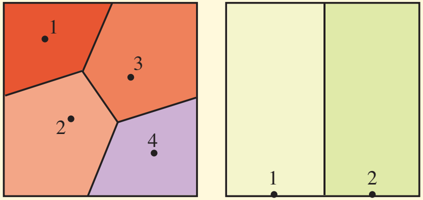
Récupération d'une fonction d'observation

- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$),

guy.cohen@mail.enpc.fr

Récupération d'une fonction d'observation

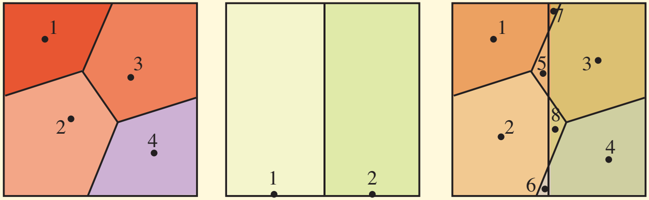
- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$), on n'a pas (forcément) $\widetilde{y} \preceq \widetilde{\xi}$.



guy.cohen@mail.enpc.fr

Récupération d'une fonction d'observation

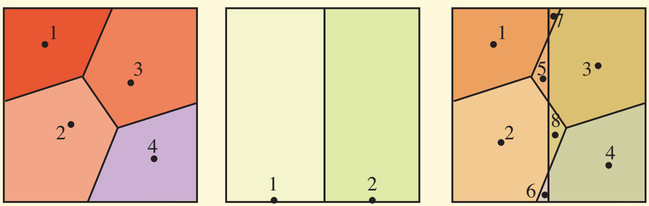
- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$), on n'a pas (forcément) $\widetilde{y} \preceq \widetilde{\xi}$.
- On peut alors remplacer $\widetilde{\xi}$ par $\widehat{\xi} \sim \widetilde{y} \vee \widetilde{\xi}$ (ici, il faut définir un codebook à 8 éléments).



guy.cohen@mail.enpc.fr

Récupération d’une fonction d’observation

- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$), on n’a pas (forcément) $\widetilde{y} \preceq \widetilde{\xi}$.
- On peut alors remplacer $\widetilde{\xi}$ par $\widehat{\xi} \sim \widetilde{y} \vee \widetilde{\xi}$ (ici, il faut définir un codebook à 8 éléments).
- Alors $\widetilde{y} \preceq \widehat{\xi}$

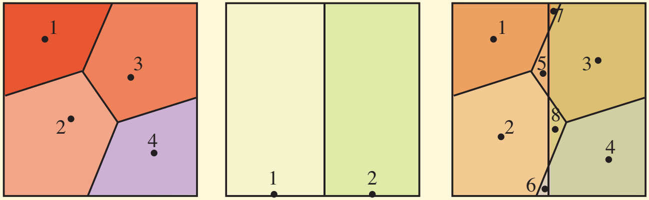


Introduction
Typologie des . . .
Un exemple en . . .
Aperçu succinct des . . .
Treillis des partitions . . .
Discrétisation, . . .
Résultats . . .
Conditions . . .
Page 23 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

Récupération d'une fonction d'observation

- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$), on n'a pas (forcément) $\widetilde{y} \preceq \widetilde{\xi}$.
- On peut alors remplacer $\widetilde{\xi}$ par $\widehat{\xi} \sim \widetilde{y} \vee \widetilde{\xi}$ (ici, il faut définir un codebook à 8 éléments).
- Alors $\widetilde{y} \preceq \widehat{\xi} \iff \exists \widetilde{h} : \widetilde{y} = \widetilde{h}(\widehat{\xi})$

	$\widehat{\xi}_1$	$\widehat{\xi}_2$	$\widehat{\xi}_3$	$\widehat{\xi}_4$	$\widehat{\xi}_5$	$\widehat{\xi}_6$	$\widehat{\xi}_7$	$\widehat{\xi}_8$
$\widetilde{y}$	1	1	2	2	1	1	2	2



guy.cohen@mail.enpc.fr

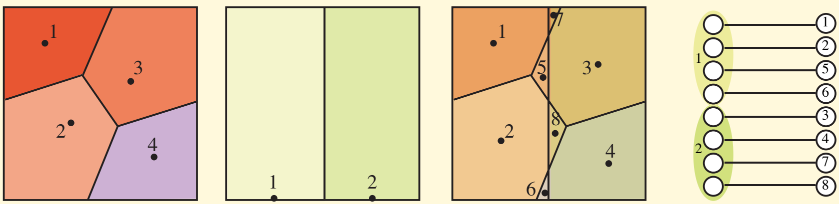
Récupération d'une fonction d'observation

- Observons que, bien que $y \preceq \xi$ ($y = h(\xi)$), on n'a pas (forcément) $\widetilde{y} \preceq \widetilde{\xi}$.
- On peut alors remplacer $\widetilde{\xi}$ par $\widehat{\xi} \sim \widetilde{y} \vee \widetilde{\xi}$ (ici, il faut définir un codebook à 8 éléments).
- Alors $\widetilde{y} \preceq \widehat{\xi} \iff \exists \widetilde{h} : \widetilde{y} = \widetilde{h}(\widehat{\xi})$

	$\widehat{\xi}_1$	$\widehat{\xi}_2$	$\widehat{\xi}_3$	$\widehat{\xi}_4$	$\widehat{\xi}_5$	$\widehat{\xi}_6$	$\widehat{\xi}_7$	$\widehat{\xi}_8$
$\widetilde{y}$	1	1	2	2	1	1	2	2

- On doit maintenant résoudre

$$\min_{u_1,u_2} \left(\sum_{j=1}^{8} \mathcal{P}(\Omega_j) \times J\left(u_{\widetilde{h}(\widehat{\xi}_j)}, \widehat{\xi}_j\right) \right) .$$



Page 23 sur 33
Plein écran
Zoom
Main
Copier
Coller
Signets
Dest.

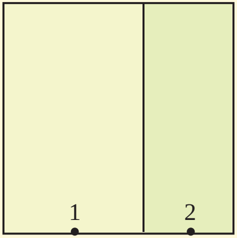
guy.cohen@mail.enpc.fr

Une façon de préserver des “arbres de scénarios”

Plein écran
Zoom Main
Copier Coller
Signets Dest.

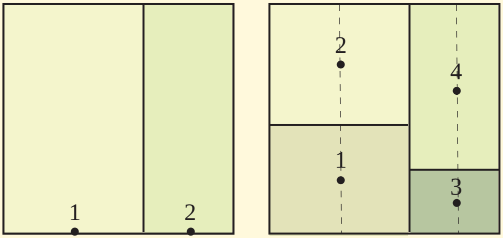
guy.cohen@mail.enpc.fr

Une façon de préserver des “arbres de scénarios”



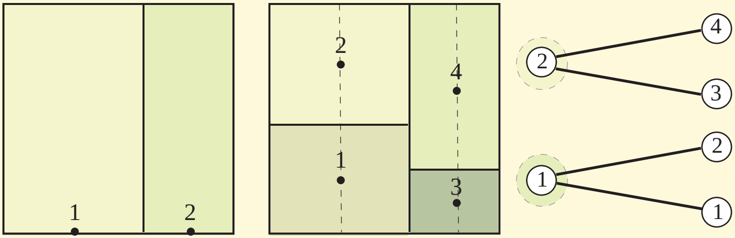
guy.cohen@mail.enpc.fr

Une façon de préserver des “arbres de scénarios”



guy.cohen@mail.enpc.fr

Une façon de préserver des “arbres de scénarios”



guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

Problème de commande optimale stochastique en temps discret avec

guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

Problème de commande optimale stochastique en temps discret avec

- une variable d'état (niveau d'eau dans un barrage entre 0 et 2),

guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

Problème de commande optimale stochastique en temps discret avec

- une variable d'état (niveau d'eau dans un barrage entre 0 et 2),
- une variable de commande (turbiné entre 0 et 1),

guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

Problème de commande optimale stochastique en temps discret avec

- une variable d'état (niveau d'eau dans un barrage entre 0 et 2),
- une variable de commande (turbiné entre 0 et 1),
- deux bruits blancs (demande d'électricité et apport d'eau) observés de façon causale,

guy.cohen@mail.enpc.fr

7. Résultats numériques sur un problème test

7.1. Caractéristiques du problème test

Problème de commande optimale stochastique en temps discret avec

- une variable d'état (niveau d'eau dans un barrage entre 0 et 2),
- une variable de commande (turbiné entre 0 et 1),
- deux bruits blancs (demande d'électricité et apport d'eau) observés de façon causale,
- un coût à optimiser (avec ressource auxiliaire à prix marginal variable) sur 24 pas de temps.

guy.cohen@mail.enpc.fr

7.2. Examen des résultats numériques

guy.cohen@mail.enpc.fr

MODE 2004
Le Havre
25-27 mars

7.2. Examen des résultats numériques

- On résout numériquement le problème sur l’arbre de scénarios obtenus à partir de 3000 chroniques par quantification “forward” pour obtenir un arbre binaire.

guy.cohen@mail.enpc.fr

7.2. Examen des résultats numériques

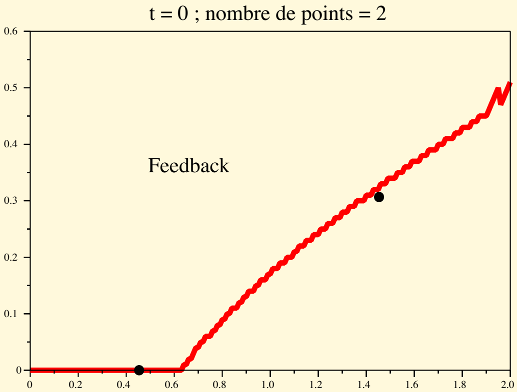
- On résout numériquement le problème sur l'arbre de scénarios obtenus à partir de 3000 chroniques par quantification "forward" pour obtenir un arbre binaire.
- La résolution donne en chaque nœud de l'arbre les valeurs optimales de
  - l'état,
  - la commande,
  - l'état adjoint (nécessaire pour calculer le gradient).

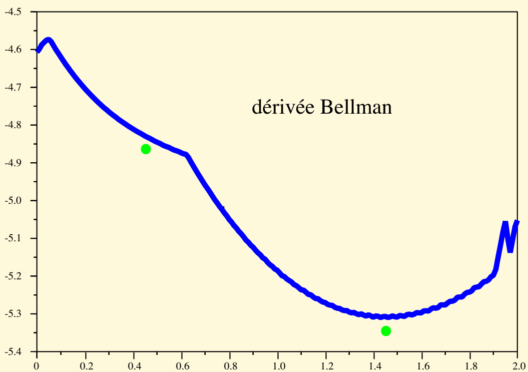
guy.cohen@mail.enpc.fr

7.2. Examen des résultats numériques

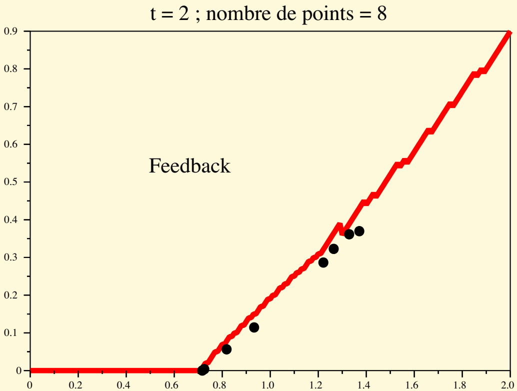
- On résout numériquement le problème sur l'arbre de scénarios obtenus à partir de 3000 chroniques par quantification "forward" pour obtenir un arbre binaire.
- La résolution donne en chaque nœud de l'arbre les valeurs optimales de
  - l'état,
  - la commande,
  - l'état adjoint (nécessaire pour calculer le gradient).
- On examine, pour tous les nœuds de l'arbre à chacun des pas de temps, le nuage de points des couples (état, commande) et (état, état adjoint).

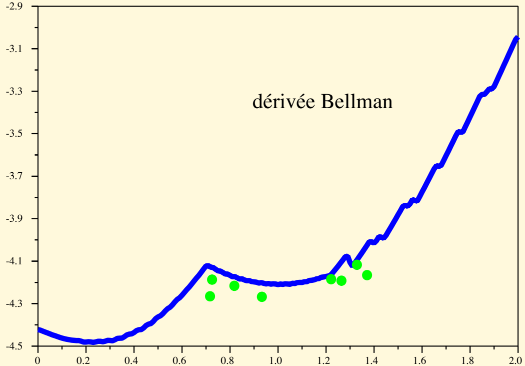
guy.cohen@mail.enpc.fr





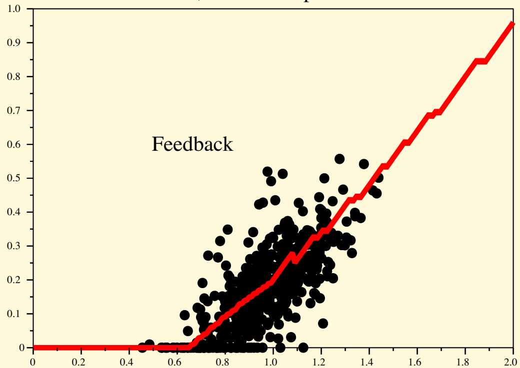
guy.cohen@mail.enpc.fr

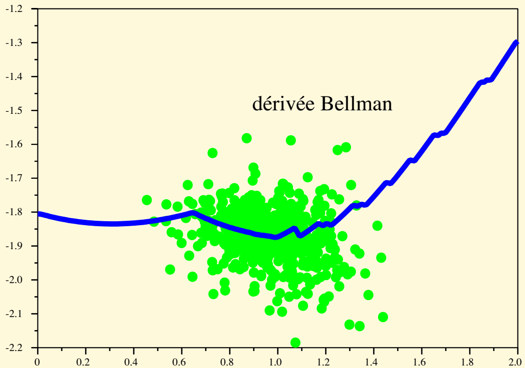




guy.cohen@mail.enpc.fr

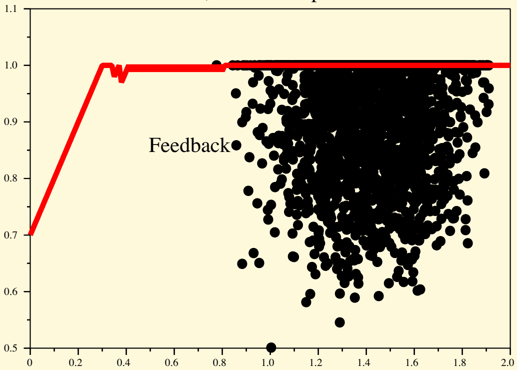
t = 8 ; nombre de points = 509

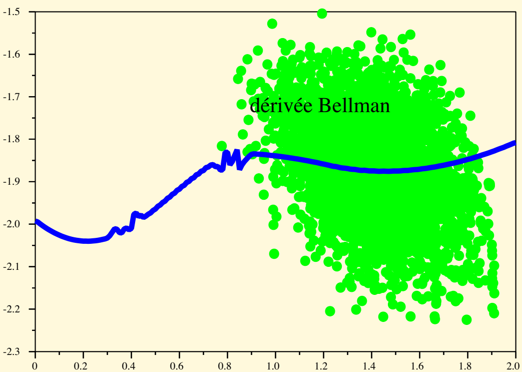




guy.cohen@mail.enpc.fr

t = 16 ; nombre de points = 3000





guy.cohen@mail.enpc.fr

8. Conditions d'optimalité de type variationnel en commande optimale stochastique

8. Conditions d’optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

guy.cohen@mail.enpc.fr

8. Conditions d'optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

Dynamique :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

où x_0 et w_t sont des variables aléatoires *pas nécessairement indépendantes*.

guy.cohen@mail.enpc.fr

8. Conditions d’optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

Dynamique :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

où x_0 et w_t sont des variables aléatoires *pas nécessairement indépendantes*.

Coût :

$$\mathbb{E}\Big(K(x_T) + \sum_{t=0}^{T-1} L_t(x_t, u_t, w_{t+1})\Big) \ .$$

guy.cohen@mail.enpc.fr

8. Conditions d'optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

Dynamique :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

où x_0 et w_t sont des variables aléatoires *pas nécessairement indépendantes*.

Coût :

$$\mathbb{E}\Big(K(x_T) + \sum_{t=0}^{T-1} L_t(x_t, u_t, w_{t+1})\Big) \ .$$

Contrainte de mesurabilité : soit $\mathcal{F}_t$ la σ-algèbre engendrée par $\{x_0, w_1, \dots, w_t\}$

guy.cohen@mail.enpc.fr

8. Conditions d'optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

Dynamique :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

où x_0 et w_t sont des variables aléatoires *pas nécessairement indépendantes*.

Coût :

$$\mathbb{E}\Big(K(x_T) + \sum_{t=0}^{T-1} L_t(x_t, u_t, w_{t+1})\Big) \ .$$

Contrainte de mesurabilité : soit $\mathcal{F}_t$ la σ-algèbre engendrée par $\{x_0, w_1, \ldots, w_t\}$: on impose à la commande u_t d'être mesurable par rapport à $\mathcal{G}_t \subset \mathcal{F}_t$

Plein écran
Zoom Main
Copier Coller
Signets Dest.

8. Conditions d’optimalité de type variationnel en commande optimale stochastique

8.1. Formulation

Dynamique :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

où x_0 et w_t sont des variables aléatoires *pas nécessairement indépendantes*.

Coût :

$$\mathbb{E}\Big(K(x_T) + \sum_{t=0}^{T-1} L_t(x_t, u_t, w_{t+1}) \Big) \ .$$

Contrainte de mesurabilité : soit $\mathcal{F}_t$ la σ-algèbre engendrée par $\{x_0, w_1, \dots, w_t\}$: on impose à la commande u_t d’être mesurable par rapport à $\mathcal{G}_t \subset \mathcal{F}_t$, ce qui est équivalent à :

$$u_t = \mathbb{E}(u_t \mid \mathcal{G}_t) \ , \quad \text{p.s.}$$

8.2. Conditions d’optimalité

guy.cohen@mail.enpc.fr

8.2. Conditions d'optimalité

Première forme :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

$$\lambda_t = \big((f_t)'_x(x_t, u_t, w_{t+1})\big)^\top \lambda_{t+1} + \big((L_t)'_x(x_t, u_t, w_{t+1})\big)^\top \ ,$$

$$\lambda_T = \big(K'(x_T)\big)^\top \ ,$$

$$\mathbb{E}\Big(\big((f_t)'_u(x_t, u_t, w_{t+1})\big)^\top \lambda_{t+1} + \big((L_t)'_u(x_t, u_t, w_{t+1})\big)^\top \ \Big|\ \mathcal{G}_t\Big) = 0 \ .$$

guy.cohen@mail.enpc.fr

8.2. Conditions d'optimalité

Première forme :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

$$\lambda_t = \big((f_t)'_x(x_t, u_t, w_{t+1})\big)^\top \lambda_{t+1} + \big((L_t)'_x(x_t, u_t, w_{t+1})\big)^\top \ ,$$

$$\lambda_T = \big(K'(x_T)\big)^\top \ ,$$

$$\mathbb{E}\Big(\big((f_t)'_u(x_t, u_t, w_{t+1})\big)^\top \lambda_{t+1} + \big((L_t)'_u(x_t, u_t, w_{t+1})\big)^\top \mid \mathcal{G}_t\Big) = 0 \ .$$

Deuxième forme : on pose $\mu_t = \mathbb{E}(\lambda_t \mid \mathcal{F}_t)$

guy.cohen@mail.enpc.fr

8.2. Conditions d'optimalité

Première forme :

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

$$\lambda_t = \Big((f_t)'_x(x_t, u_t, w_{t+1})\Big)^\top \lambda_{t+1} + \Big((L_t)'_x(x_t, u_t, w_{t+1})\Big)^\top \ ,$$

$$\lambda_T = \Big(K'(x_T)\Big)^\top \ ,$$

$$\mathbb{E}\bigg(\Big((f_t)'_u(x_t, u_t, w_{t+1})\Big)^\top \lambda_{t+1} + \Big((L_t)'_u(x_t, u_t, w_{t+1})\Big)^\top \ \Big| \ \mathcal{G}_t\bigg) = 0 \ .$$

Deuxième forme : on pose $\mu_t = \mathbb{E}(\lambda_t \mid \mathcal{F}_t)$

$$x_{t+1} = f_t(x_t, u_t, w_{t+1}) \ ,$$

$$\mu_t = \mathbb{E}\bigg(\Big((f_t)'_x(x_t, u_t, w_{t+1})\Big)^\top \mu_{t+1} + \Big((L_t)'_x(x_t, u_t, w_{t+1})\Big)^\top \ \Big| \ \mathcal{F}_t\bigg) \ ,$$

$$\mu_T = \Big(K'(x_T)\Big)^\top \ ,$$

$$\mathbb{E}\bigg(\Big((f_t)'_u(x_t, u_t, w_{t+1})\Big)^\top \mu_{t+1} + \Big((L_t)'_u(x_t, u_t, w_{t+1})\Big)^\top \ \Big| \ \mathcal{G}_t\bigg) = 0 \ .$$

guy.cohen@mail.enpc.fr

8.3. Exploitation (de la première forme) dans une méthode "particulaire" (cas $\mathcal{G}_t = \mathcal{F}_t$)

guy.cohen@mail.enpc.fr

8.3. Exploitation (de la première forme) dans une méthode "particulaire" (cas $\mathcal{G}_t = \mathcal{F}_t$)

Les "particules" sont des "scénarios"

$$(x_0^i, w_1^i, \dots, w_T^i), \quad i = 1, \dots, N.$$

guy.cohen@mail.enpc.fr

8.3. Exploitation (de la première forme) dans une méthode "particulaire" (cas $\mathcal{G}_t = \mathcal{F}_t$)

Les "particules" sont des "scénarios"

$$(x_0^i, w_1^i, \dots, w_T^i), \quad i = 1, \dots, N.$$

Pour chaque particule i, on calcule :

$$x_{t+1}^i = f_t(x_t^i, u_t^i, w_{t+1}^i) \ ,$$

$$\lambda_t^i = \big((f_t)'_x(x_t^i, u_t^i, w_{t+1}^i)\big)^\top \lambda_{t+1}^i + \big((L_t)'_x(x_t^i, u_t^i, w_{t+1}^i)\big)^\top \ ,$$

$$\lambda_T^i = \big(K'(x_T^i)\big)^\top \ ,$$

guy.cohen@mail.enpc.fr

8.3. Exploitation (de la première forme) dans une méthode “particulaire” (cas $\mathcal{G}_t = \mathcal{F}_t$)

Les “particules” sont des “scénarios”

$$(x_0^i, w_1^i, \ldots, w_T^i), \quad i = 1, \ldots, N.$$

Pour chaque particule i, on calcule :

$$x_{t+1}^i = f_t(x_t^i, u_t^i, w_{t+1}^i) \ ,$$

$$\lambda_t^i = \big((f_t)'_x(x_t^i, u_t^i, w_{t+1}^i)\big)^\top \lambda_{t+1}^i + \big((L_t)'_x(x_t^i, u_t^i, w_{t+1}^i)\big)^\top \ ,$$

$$\lambda_T^i = \big(K'(x_T^i)\big)^\top \ ,$$

et on fait ensuite interagir les particules pour évaluer le long de chaque particule le gradient projeté par rapport à U_t pour tout t :

$$\mathbb{E}\Big(\big((f_t)'_u(X_t, U_t, W_{t+1})\big)^\top \Lambda_{t+1} + \big((L_t)'_u(X_t, U_t, W_{t+1})\big)^\top \;\Big|\; (x_0^i, w_1^i, \ldots, w_t^i)\Big) \ .$$

guy.cohen@mail.enpc.fr