Direction de la Recherche

Département Informatique

Présentation

Vers une approche modulaire de la prévision

Denis BONNET
[**SNCF/ENST**]
bonnet@inf.enst.fr

Véronique PERRAULT
[**SNCF**]
perrault@dr.sncf.fr

Alain GRUMBACH
[**ENST**]
grumbach@inf.enst.fr

SNCF
Recherche Prospective
Département Génie Décisionnel
45, rue de Londres
75379 Paris CEDEX 08

ENST
Département Informatique
Groupe MILC
46, rue de Barrault
75634 Paris CEDEX 13

Plan de l'exposé

Principes

Comprendre les mécanismes fondamentaux des séries temporelles.
Comprendre la série elle-même afin d'en faciliter la prédiction.
Evaluer la qualité des résultats obtenus grâce à des critères pertinents.
Comparer les résultats obtenus par le biais des critères choisis

Statistique

Connexionnisme

Modèles auto-régressifs

Statistique	Connexionnisme
Modèles AR	Modèles NAR
Modèles TAR	Modèles NTAR

Modèles récurrents

Statistique	Connexionnisme
Modèles ARMA	Modèles NARMA
Modèles ARIMA	Modèles NARMA
Modèles ARARMA	Modèles δ-NARMA

Hybridation symboliques

Choisir une manière de **représenter** les variables symboliques.
Comprendre quelles sont les variables **pertinentes**.
Intégrer les variables **symboliques** aux modèles **univariés**.
Evaluer la capacité des modèles univariés à s'affranchir des variables symboliques.

Principes

Comprendre

Comprendre les mécanismes fondamentaux des séries temporelles.

Comprendre la série elle-même afin d'en faciliter la prédiction.

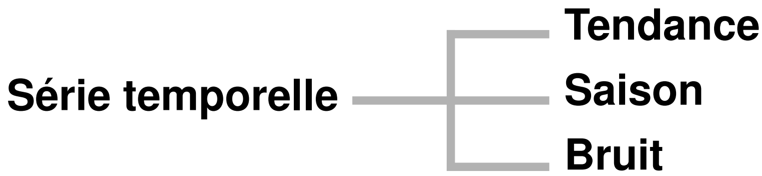
Prédire

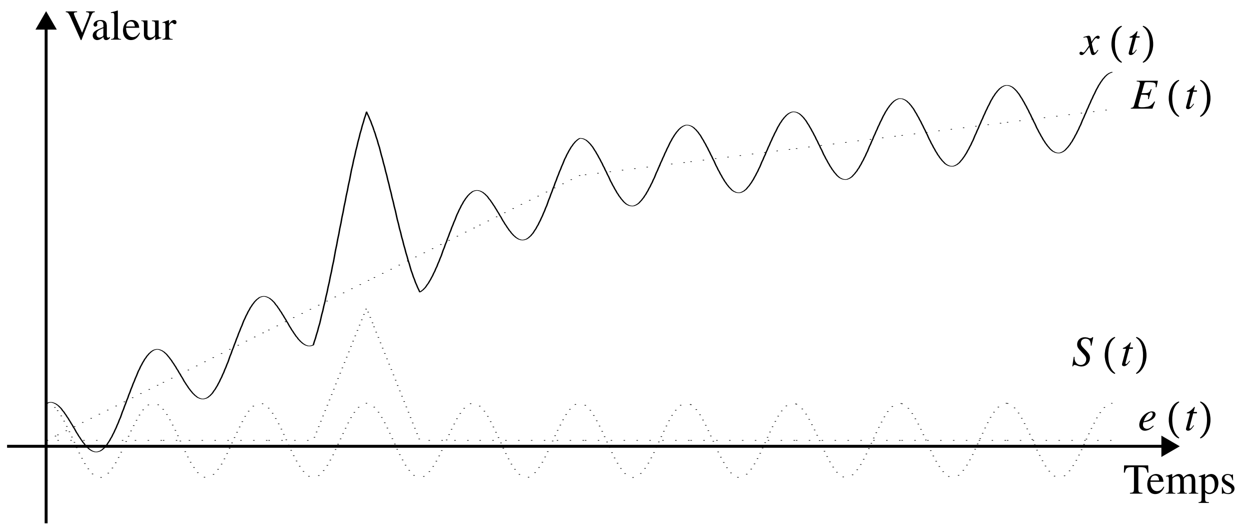
Evaluer

Evaluer la qualité des résultats obtenus grâce à des critères pertinents.

Comparer les résultats obtenus par le biais des critères choisis.

Comprendre les mécanismes fondamentaux des séries temporelles





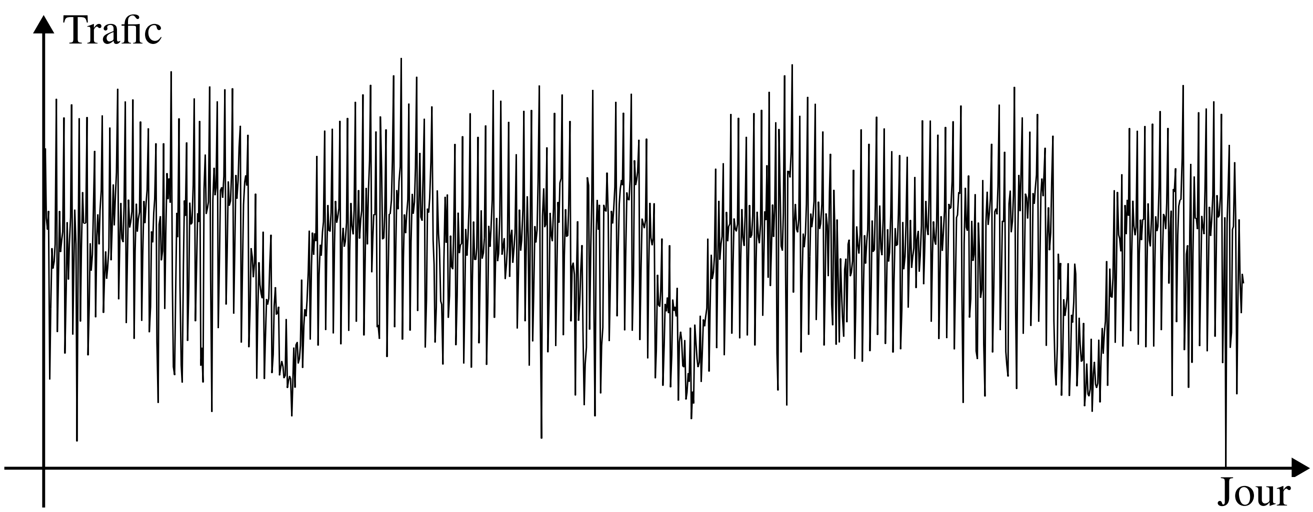
$$x(t) = E(t) + S(t) + e(t)$$

ou

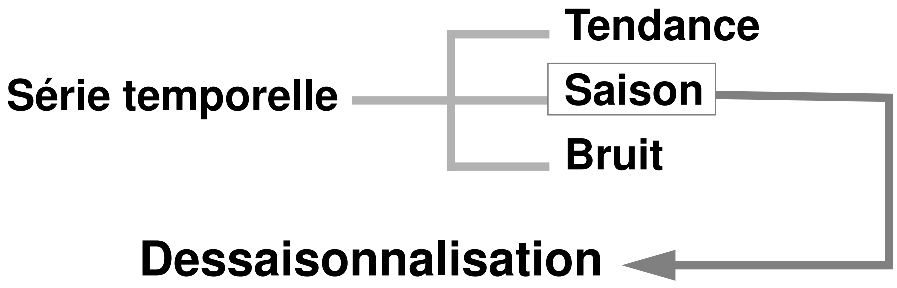
$$x(t) = E(t) \cdot S(t) \cdot [1 + e(t)]$$

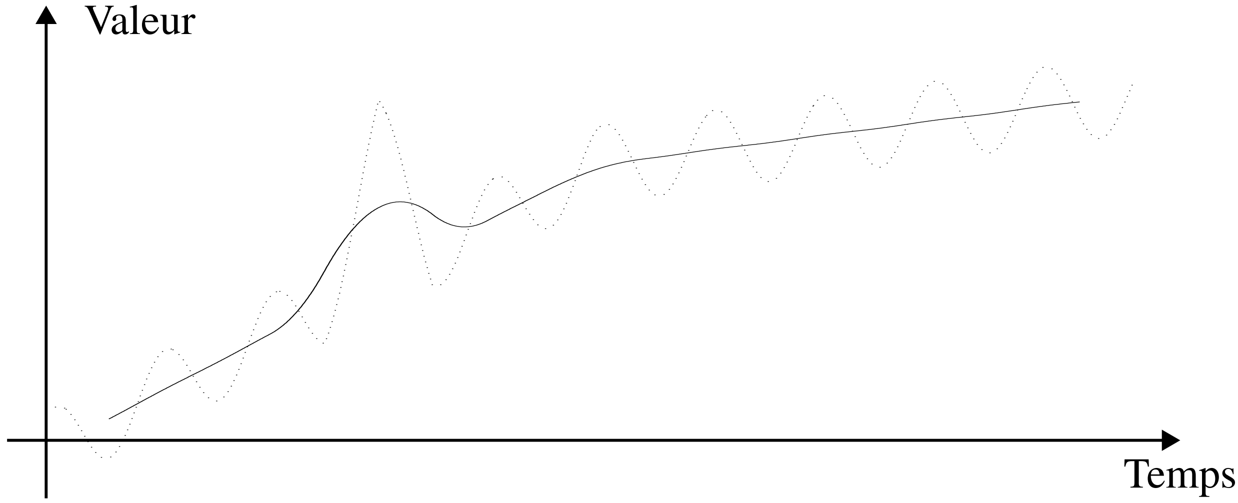
E X E M P L E

Le trafic voyageur TGV Paris-Lyon journalier



Comprendre la série elle-même afin d'en faciliter la prédiction (1)

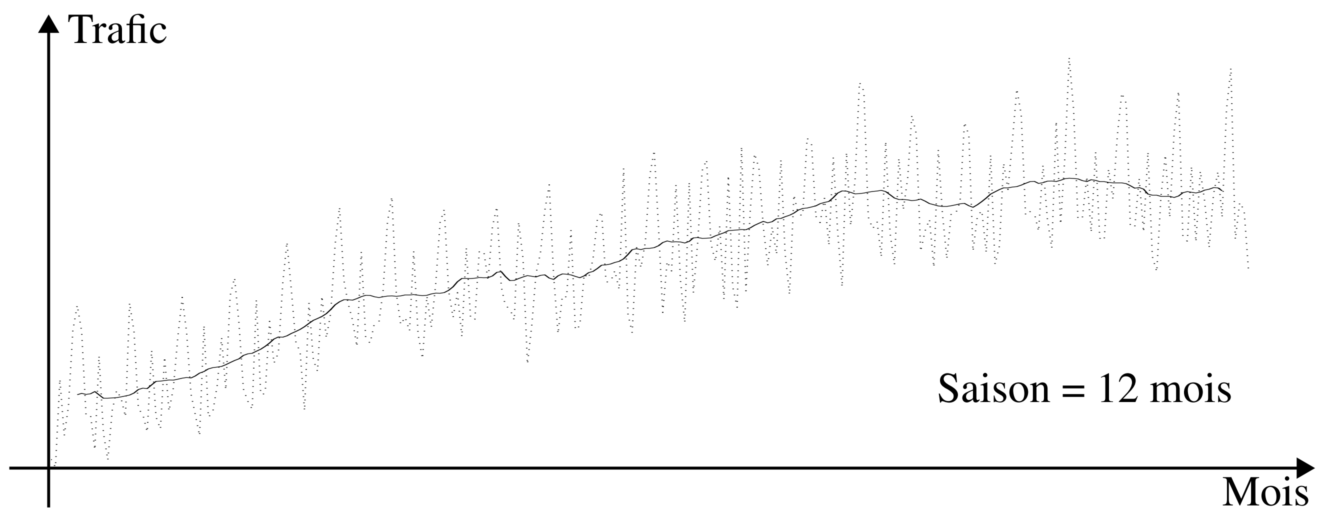




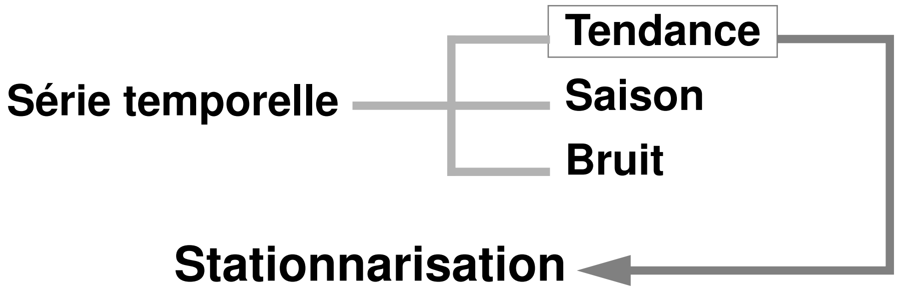
La dessaisonnalisation se fait en général par la méthode des **moyennes mobiles** :
Principe : la partie périodique a une **moyenne constante** sur une **période**.
Propriété : les moyennes mobiles sont des **filtres passe-bas**.
Défauts : les moyennes mobiles **noient le bruit** dans la tendance et il est difficile de **dessaisonnaliser** la **fin de la série**.

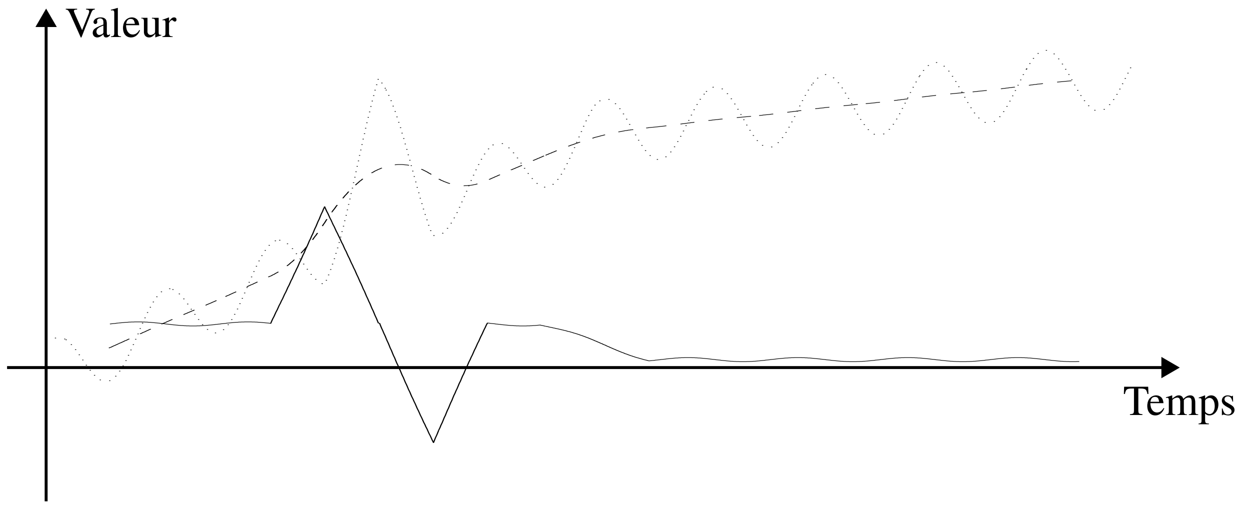
E X E M P L E

Le trafic voyageur national mensuel



Comprendre la série elle-même afin d'en faciliter la prédiction (2)



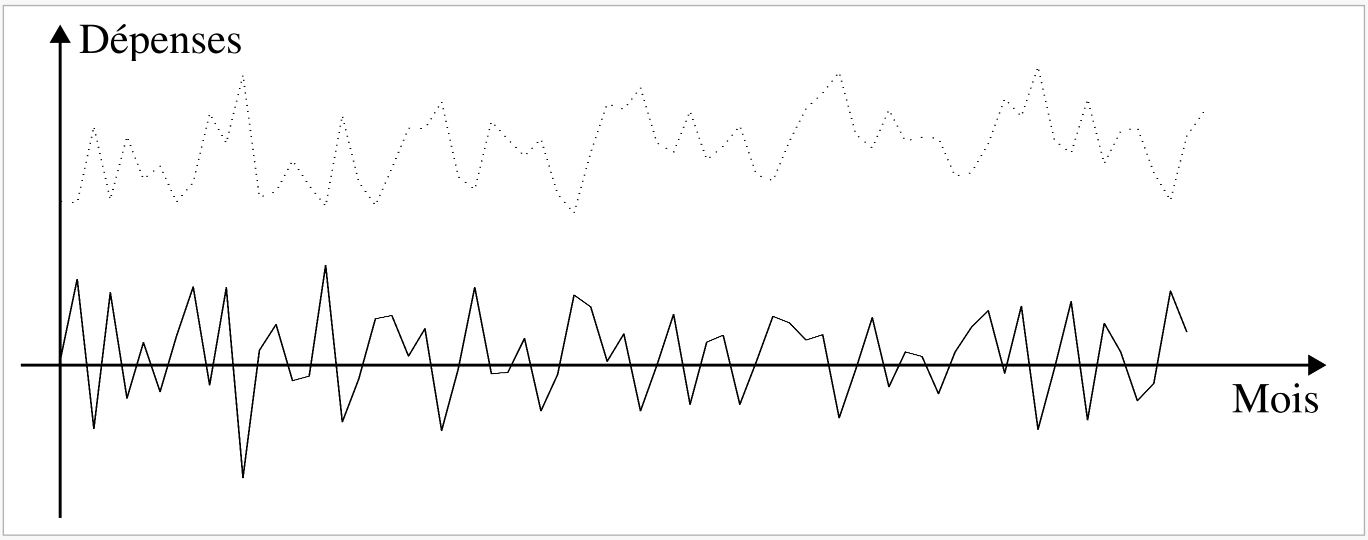


La stationnarisation se fait en général par la méthode de la **différenciation**.

Principe : une des **dérivées** du signal est en général **stationnaire**.
Propriété : la différenciation est un **filtre passe-haut**.
Défauts : la différenciation intègre les **changements de tendance** dans le bruit.

E X E M P L E

Dépenses mensuelles d'une ligne de budget



Evaluer la qualité des résultats obtenus grâce à des critères pertinents

Problème : nous essayons d'**évaluer** les prévision faites sur des **séries différentes** par des **modèles différents**.

Problème-indépendance → Insensibilité à la longueur de la série. Insensibilité à ses moments d'ordre 1 et 2.

Modèle-indépendance → Insensibilité aux hypothèses du modèle. Insensibilité aux paramètres du modèle.

SE

(erreur quadratique moyenne)

Insensible aux paramètres et à la moyenne (moment d'ordre 1).

$$SE = \sum_{t=1}^{N} [x(t) - \hat{x}(t)]^2$$

MSE

(erreur quadratique moyenne)

SE + Insensible à la longueur la série.

$$MSE = \frac{SE}{N}$$

ARV

(variance relative moyenne)

MSE + Insensible au moment d'ordre 2.

$$ARV = \frac{MSE}{\delta^2}$$

MAPE

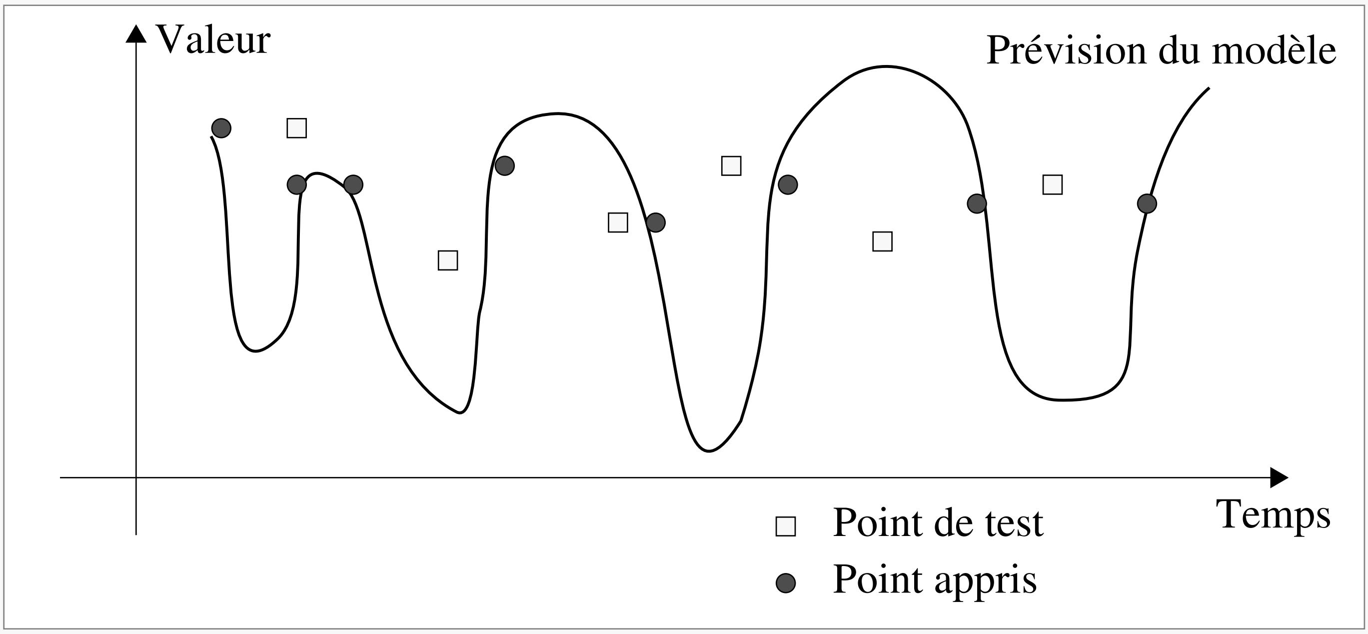
(pourcentage d'erreur absolu moyen)

idem MSE avec l'erreur absolue moyenne.

Comparer les résultats obtenus par le biais des critères choisis

PROCESSUS STOCHASTIQUES

Problème : sur-ajustement (ou sur-apprentissage) :



Solution : **Comparer** les erreurs commises sur les points **appris** et sur les points de **test**.
Ajouter un **bruit variable** sur les données pour empécher le sur-ajustement (valable si l'ajustement est itératif).

Apprentissage **implicite** sur les points de test.

Nécessité de définir un **troisième ensemble** (dit de **validation**) permettant de procéder à une **comparaison objective** des modèles et qui n'est dévoilé qu'au moment de la comparaison finale des modèles.

Comme les points appris et testés sont choisis **aléatoirement** (*idem* pour les points de validation), il est nécessaire de procéder à un **nombre important de retirages** pour conférer une **validité statistique** à la comparaison.

Conclusion

Pré-traitements :

La nature du problème à traiter (la prévision de séries temporelles) permet de procéder à certains **pré-traitements** permettant de faciliter la tâche du prédicteur. Ces pré-traitements ont certes des **avantages**, mais **modifient** profondément la **nature de la série**.

Par soucis de **généricité** et afin d'**éviter ce biais**, nous avons choisi pour nos modèles connexionnistes de ne procéder à **aucun pré-traitement**, laissant aux modèles le soin d'effectuer ceux qui semblent pertinents.

Evaluation :

Etant donnés les pièges d'une évaluation partiale, nous avons choisi les **critères** qui nous paraissaient les plus pertinents en insistant sur la **validité statistique** des résultats obtenus. Cette validité nécessite un **grand nombre de simulations** (de l'ordre de 100 par modèle) et donc des moyens de calculs importants (15 jours de simulation pour les modèles les plus complexes). Cependant, en phase d'exploitation, une telle validation n'est plus nécessaire.

Nous avons testé selon ces techniques tous les modèles indiqués par la suite et nous avons pu **comparer** leur résultats sur les **trois séries** évoquées précédemment.

Auto-régressifs

Comprendre

Prédire

- **Evaluer** la dimension auto-régressive d'une série.
- Modèles **statistisques** (AR, TAR).
- Modèles **connexionnistes** (NAR, NTAR).

Evaluer

Evaluer la dimension auto-régressive d'une série

DEFINITION

Un **processus auto-régressif** général s'exprime sous la forme :

$$x(t) = f[x(t-1), \ldots, x(t-n)] + e(t)$$

où $e(t)$ est un bruit gaussien.

On en déduit le **prédicteur auto-régressif** associé :

$$\hat{x}(t) = f[x(t-1), \ldots, x(t-n)]$$

Problème : évaluer la valeur de n.

THEORIE DU CHAOS

Principe : recherche d'un **hyperplan** dans lequel la représentation de $x(t)$ en fonction des $x(t\text{-}1),\ldots,x(t\text{-}n)$ est une **application**.

Algorithme : calcul de la **dimension fractale** de l'**attracteur étrange** associé à la série par la méthode de Grassberger et Proccacia.

Inconvénient : cette méthode ne fonctionne que si la série est **chaotique** et il est très difficile de savoir si une série est chaotique ou non.

CONNEXIONNISME

Principe : trouver la dimension auto-régressive donnant les **meilleures performances** en terme de prévision.

Algorithme : il s'agit de choisir un modèle susceptible de **modéliser un chaos** (par exemple un Perceptron Multi-couches). Il faut alors s'assurer que l'on n'effectue **aucun sur-apprentisage** et on cherche alors le minimum de l'ARV en fonction de n.

Inconvénient : il est extrêmement difficile d'éviter tout sur-apprentissage.

Le modèle statistique AR

DEFINITION

Un **processus AR(n)** général s'exprime sous la forme :

$$x(t) = \alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_n \cdot x(t-n) + e(t)$$

où $e(t)$ est un bruit gaussien.

On en déduit le **prédicteur AR(n)** associé :

$$\hat{x}(t) = \alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_n \cdot x(t-n)$$

f est donc la fonction de transfert simple :

$$f(x_0, \ldots, x_n) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \cdot x_i$$

Propriétés : Le prédicteur AR est **linéaire**.

Le prédicteur AR est très **simple** (n paramètres).

Le calcul des paramètres est très facile et **peu coûteux en temps de calcul** (méthode des moindres carrés par exemple).

Applications : Les séries **très simples** (quasi-linéaires).

Les séries **très complexes** où la meilleure politique est de **reproduire** à l'instant t la valeur observée à l'instant t-1.

Inconvénients: Il leur est impossible de prédire des séries **non-linéaires**, même simples.

Le modèle connexionniste NAR (1)

D E F I N I T I O N

Un **processus NAR(*n*)** général s'exprime sous la forme :

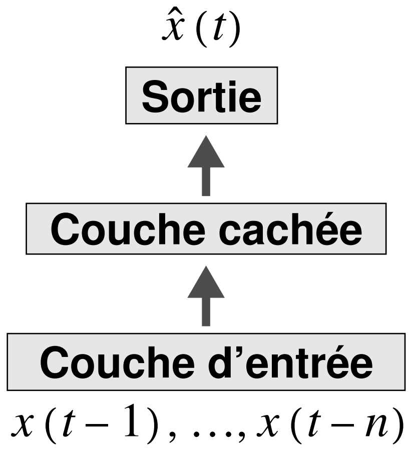
$$x(t) = f[x(t-1), \ldots, x(t-n)] + e(t)$$

où $e(t)$ est un bruit gaussien et f la fonction de transfert d'un Perceptron Multi-couches.

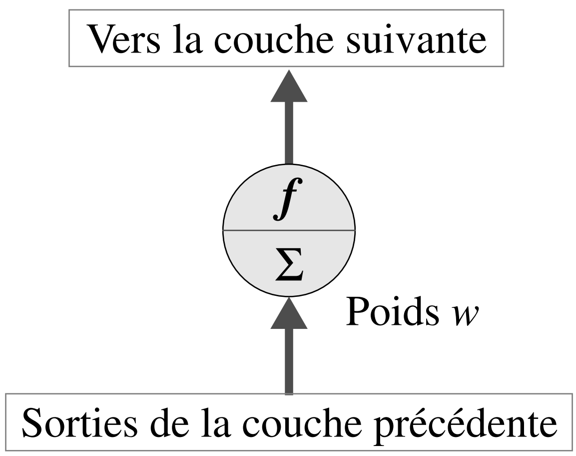
On en déduit le **prédicteur NAR(*n*)** associé :

$$\hat{x}(t) = f[x(t-1), \ldots, x(t-n)]$$

Principe d'un Perceptron Multicouche (MLP)

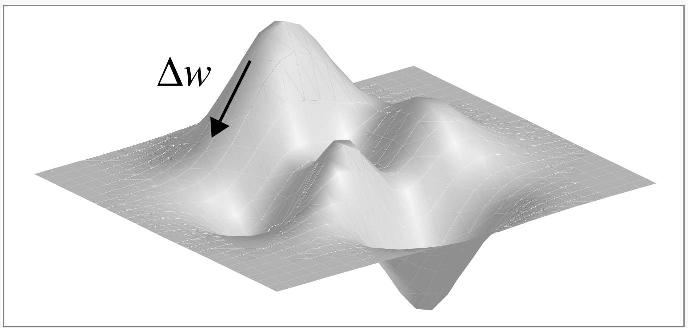


Principe d'un neurone



Le modèle connexionniste NAR (2)

L A   D E S C E N T E   D E   G R A D I E N T



On **modifie les poids** d'une quantité Δw telle que les nouveaux poids produisent une **erreur plus faible** que les ancens. Si on écrit le **développement de Taylor** de l'erreur E en fonction de w, on trouve que Δw=-ν grad E est une bonne approximation.

Dans le cas d'un MLP, la descente de gradient s'exprime simplement sous la forme d'une **formule récurrente** et permet d'ajuster les poids simplement.

Problèmes :

Cette méthode n'est **pas optimale** en terme de **temps de calcul**.
Elle ne permet pas de s'affranchir du problème des **minima locaux**.

Algorithme d'apprentissage :

`Tant que` l'erreur quadratique moyenne est trop élevée
  `Pour t dans` l'ensemble des points à apprendre
    Présentation de l'exemple `t` et calcul de la prévision de x(`t`).
    Calcul de l'erreur associée.
    Rétropropagation de l'erreur et modification des poids.
  `fin`
`fin`

Le modèle connexionniste NAR (3)

Propriétés : Le prédicteur NAR est **non-linéaire**.

Le prédicteur NAR est un **approximateur universel**.

Le prédicteur NAR est capable de prédire un **chaos déterministe**.

La complexité du prédicteur NAR est **ajustable** en fonction de celle supposée de la série à prédire.

Le calcul des paramètres est **automatique**.

Applications : Virtuellement tous les processus auto-régressif.

Inconvénients: Il est très **délicat** d'éviter tout **sur-apprentissage**.

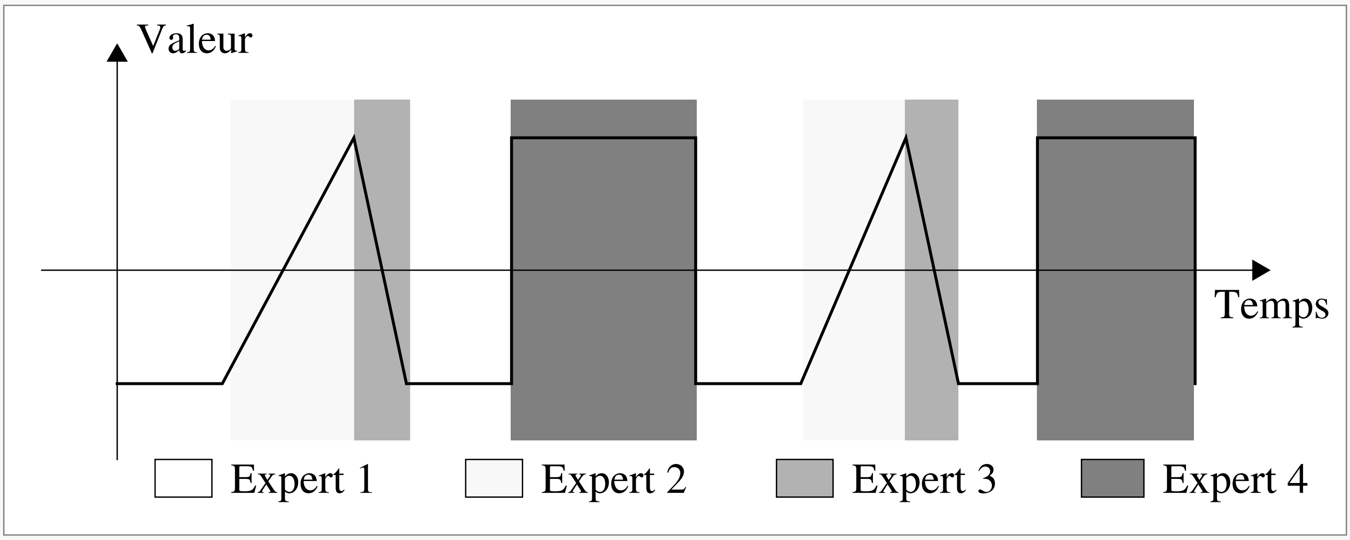
L'a**justement des paramètres** du système (nombre de neurones par couche) est **difficile** (mais il existe des méthodes d'élagage permettant de s'en affranchir : weight-decay, weight-elimination...).

Le **temps de calcul** est **important** par rapport à celui d'un modèle AR.

Dans le cas où l'on utilise la rétropropagation du gradient, il est **difficile** de **juger** de la **qualité** de la **convergence** du modèle (risque de tomber dans un minimum local).

Principe des modèles TAR et NTAR

D I V I S E R   P O U R   R E G N E R



Deux approches :

Considérer **un seul modèle** prévoyant toute la série (approche AR et NAR).
Considérer **plusieurs modèles** se partageant la tâche (approche TAR et NTAR).

Approche TAR et NTAR : Nécessité de définir un **critère** (c'est à dire une fonction du temps) permettant de choisir lequel des modèles appliquer à chaque instant.

S T A T I S T I Q U E
(TAR)

Choix *a priori* du critère permettant de choisir l'expert. La **segmentation** intervient ainsi aux **endroits souhaités** par le prévisionniste.

C O N N E X I O N N I S M E
(NTAR)

Le choix du domaine de définition de chaque expert n'est pas fixé *a priori* mais au cours de l'apprentissage. Ces **domaines** ne **correspondent pas** nécessairement à ceux **envisagés** par le prévisionniste.

Le modèle statistique TAR

DEFINITION

Un processus **TAR** est un processus **AR** par morceaux :

$$x(t) = x_{i(t)}(t)$$
$$i(t) = g[x(t-1), \ldots, x(t-n)] \in \{1, \ldots, q\}$$
$$x_j(t) = \alpha_{0,j} + \alpha_{1,j}x(t-1) + \ldots + \alpha_{n_j,j}x(t-n_j) + e_j(t)$$

On en déduit le **prédicteur TAR** associé :

$$\hat{x}(t) = \hat{x}_{i(t)}(t)$$
$$i(t) = g[x(t-1), \ldots, x(t-n)] \in \{1, \ldots, q\}$$
$$\hat{x}_j(t) = \alpha_{0,j} + \alpha_{1,j}x(t-1) + \ldots + \alpha_{n_j,j}x(t-n_j)$$

Propriétés : Le prédicteur TAR est **simple** (peu de paramètres).
Le calcul des paramètres est facile et **peu coûteux en temps de calcul** (méthode des moindres carrés par exemple).
Le prédicteur AR est **linéaire par morceaux**.

Applications : Les séries **non-linéaire simples**.
Les séries à **changement de régimes**.

Inconvénients: La scission entre modèles s'effectue en **fixant *a priori*** la fonction ***g***. Ceci nuit à la généricité du modèle.

Le modèle connexionniste NTAR (1)

DEFINITION

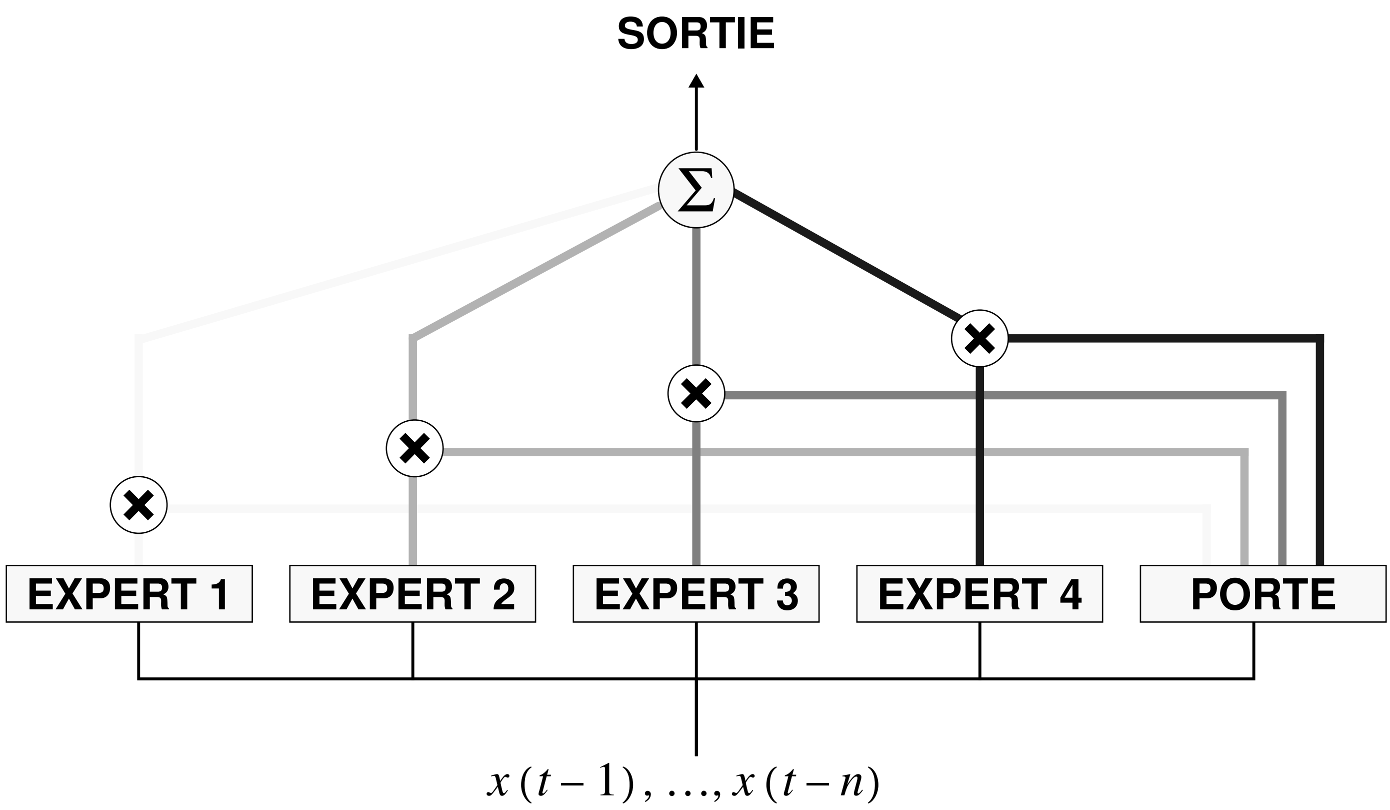
Un processus **NTAR** est un processus **AR** par morceaux :

$$x(t) = x_{i(t)}(t)$$
$$i(t) = g[x(t-1), \ldots, x(t-n)] \in \{1, \ldots, q\}$$
$$x_j(t) = \alpha_{0,j} + \alpha_{1,j}x(t-1) + \ldots + \alpha_{n_j,j}x(t-n_j) + e_j(t)$$

où g est la fonction de transfert d'un MLP.

On en déduit le **prédicteur NTAR** associé :

$$\hat{x}(t) = \hat{x}_{i(t)}(t)$$
$$i(t) = g[x(t-1), \ldots, x(t-n)] \in \{1, \ldots, q\}$$
$$\hat{x}_j(t) = \alpha_{0,j} + \alpha_{1,j}x(t-1) + \ldots + \alpha_{n_j,j}x(t-n_j)$$



Principe du modèle NTAR le plus simple (mixture of experts ou mélange d'experts).

Le modèle connexionniste NTAR (2)

APPRENTISSAGE DE LA SEGMENTATION

Principe :

L'exemple *t* est présenté et la **sortie** de chacun des **experts** est calculée.

On **compare** la sortie de chacun des experts à celle qui a réellement été observée.

La porte **renforce** la sortie assocée à chacun des experts d'une quantité **inversement proportionnelle à l'erreur** commise (valeur de 1 si l'erreur est nulle, valeur de 0 si l'erreur vaut 1).

On recommence.

Extension :

Ce point de vue intuitif peut être rendu plus rigoureux en utilisant les **probabilités conditionnelles**. Les sorties de la porte sont alors considérées comme les **probabilité *a priori*** que l'expert associé ait raison, les quantité apprise sont alors les **probabilités *a posteriori*** (c'est à dire connaissant la sortie réelle), déduites simplement par la **règle de Bayes**.

Propriétés : **Aucune hypothèse** n'est faite quant au choix des **endroits** où s'effectuent les **basculements** entre modèles.

Les experts doivent cependant être **linéaires**.

La porte doit être un **Perceptron simple** (MLP sans couche cachée).

Applications : Les mêmes que celle des modèles TAR.

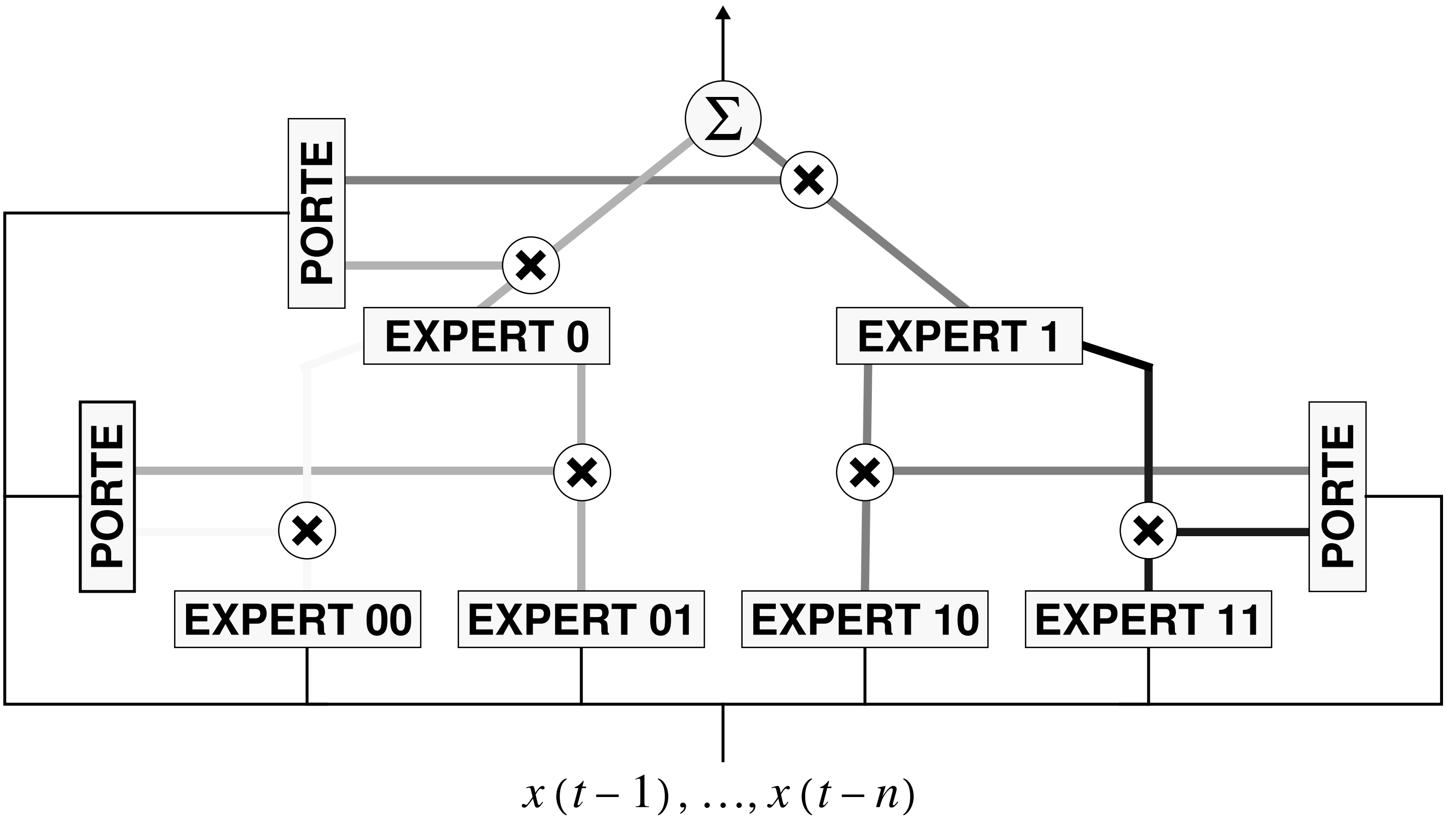
Inconvénients : Il est difficile de gérer l'**ajustement simultané des experts** (apprentissage de la série) et **des portes** (segmentation).

Il est très courant (lorsque le système a convergé) qu'**un seul** des **experts** soit **prédominant** sur toute la série.

Le modèle connexionniste NTAR (3)

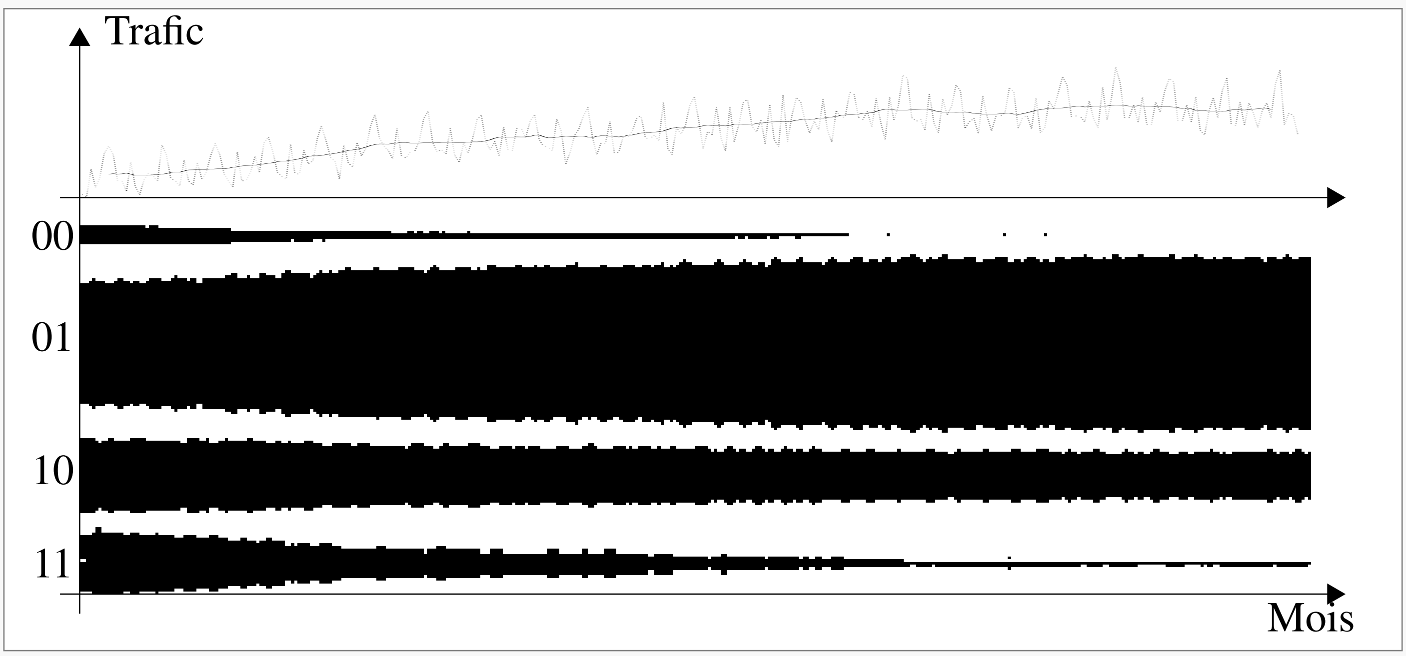
Extension :

Les **mélanges hiérarchiques d'experts** sont une extension des mélanges d'experts où les experts sont groupés par **paire**.



EXEMPLE

Segmentation du trafic mensuel



Conclusion

Nous avons pu étudier les modèles statistiques auto-régressifs les plus usités (AR et TAR) et décrire leur équivalent dans le formalisme connexionniste (NAR et NTAR).

Avantages :

Il suffit de connaître la **dimension auto-régressive** du modèle pour construire un prédicteur simple.

Ces modèles permettent de bien comprendre les **effets de retards** propres aux séries manipulées.

Inconvénients :

La **dimension auto-régressive** peut être **trop importante** (comparée au nombre de points disponibles). Dans le cas d'une série journalière par exemple, il faudrait une dimension auto-régressive de plus de 365 pour intégrer les effets annuels.

Certains processus ne **peuvent être modélisé** par des modèles **auto-régressifs**.

Ces modèles **traitent le temps** de façon **artificielle** (le temps est considéré comme une variable spatiale).

Performances :

Dans le cas des trois séries que nous avons présentées, les résultats sont les suivants en terme d'**ARV** sur les points de **validation** moyennée sur les **trois séries** (nous ne présentons pas le MAPE car ses résultats sont cohérents avec ceux de l'ARV).

AR:	1.0
NAR:	0.97
TAR:	0.94
NTAR:	0.97

Le **meilleur modèle auto-régressif** est donc le modèle statistique **TAR**, mais c'est aussi celui qui a demandé le plus de temps à développer (les basculements ont été choisis de telle sorte qu'un modèle se spécialise sur les fins de semaine, un autre sur les vacances et un dernier sur les autres jours).

Récurrents

Comprendre

Prédire

Modèles **statistiques** à **moyennes mobiles** (ARMA, ARIMA, ARARMA).

Modèles **connexionnistes récurrents** (globalement et localement).

Evaluer

Le modèle statistique ARMA

D E F I N I T I O N

Un **processus MA(n)** général s'exprime sous la forme :

$$x(t) = \beta_1 \cdot e(t-1) + \ldots + \beta_n \cdot e(t-n) + e(t)$$

où $e(t)$ est un bruit gaussien.

On en déduit le **prédicteur MA(n)** associé :

$$\hat{x}(t) = \beta_1 \cdot e(t-1) + \ldots + \beta_n \cdot e(t-n)$$

Propriétés : Un processus MA est une **suite de chocs aléatoires** gaussiens. On peut **définir** ces chocs à partir de l'**erreur commise** par le prédicteur :

$$e(t) = x(t) - \hat{x}(t)$$

Les processus MA sont **stationnaires**.

Applications : Certaines séries très **bruitées**.

Inconvénients : La **classe** des processus MA est très **restreinte**. L'**ajustement** des paramètres est **complexe** et délicat (phénomène d'oscillation).

L'utilité essentielle de ces modèles est de permettre de présenter simplement la classe des **processus ARMA** et le prédicteur associé. Ils sont quant à eux **très rarement utilisés**.

Le modèle statistique ARMA

DEFINITION

Un **processus ARMA(p,q)** général s'exprime sous la forme :

$$x(t) = \alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_p \cdot x(t-p) + \beta_1 \cdot e(t-1) + \ldots + \beta_q \cdot e(t-q) + e(t)$$

où $e(t)$ est un bruit gaussien.

On en déduit le **prédicteur ARMA(p,q)** associé :

$$\hat{x}(t) = \alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_p \cdot x(t-p) + \beta_1 \cdot e(t-1) + \ldots + \beta_q \cdot e(t-q)$$

Propriétés : Les processus ARMA sont donc la **superposition** d'un processus **AR** et d'un processus **MA**. Ils cumulent donc leurs avantages et leurs inconvénients.

Les processus ARMA sont **stationnaires**.

Un **théorème d'optimalité** indique que **tout processus stationnaire** peut être **approché** uniformément (au sens de la norme euclidienne) par un **processus ARMA**.

Il est aisé de tirer des **renseignements** sur la nature de la série à partir du processus ARMA associé.

Apprentissage **peu coûteux** en temps de **calcul**.

Applications : Virtuellement **toutes** les séries **stationnaires** (pratiquement, certaines séries fortement **non-linéaires** ne peuvent être modélisées par un tel processus).

Inconvénients: Nécessité de la **stationnarité** (au moins au second ordre).

Disposant d'un **nombre fini** de point d'apprentissage, le **théorème d'optimalité** est **sans objet**.

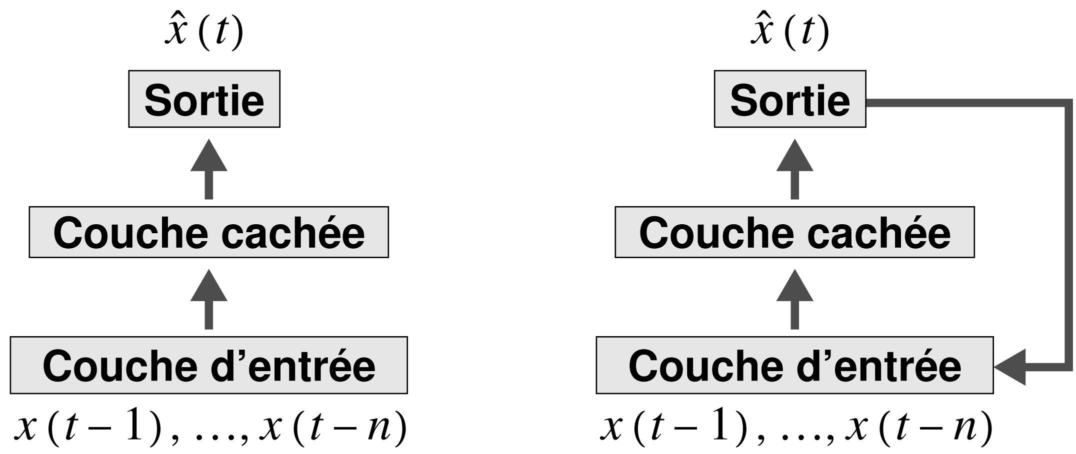
Linéarité.

Les processus ARMA sont à la bas de la statistique moderne. C'est sur eux qu'est basée toute la méthodologie de Box et Jenkins.

Le modèle connexionniste NARMA (1)

D E F I N I T I O N

Nous nommerons par la suite réseau **récurrent** tout réseau connexionniste inspiré du Perceptron Multi-couche mais **non-strictement feed-forward**.



Exemple de réseau feed-forward   Exemple de réseau récurrent

Toutes les liaisons récurrentes induisent un retard puisque sans ce retard, un phénomène de bouclage interviendrait.

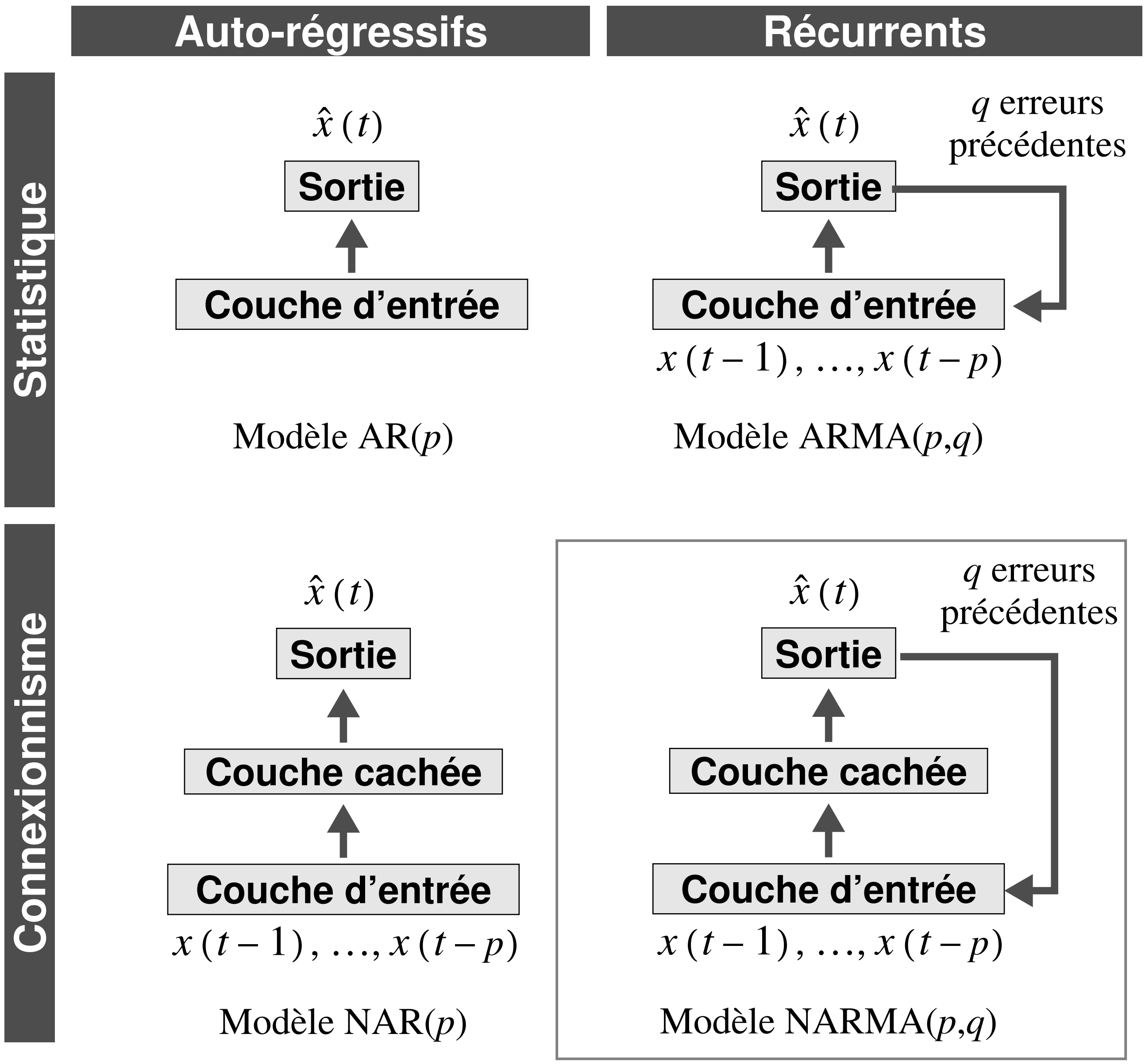
Apprentissage : Les algorithmes utilisés sont tous **basés sur la rétropropagation du gradient** (Back-Propagation Trough Time ou Real Time Recurrent Learning).

Quelques exemples :
Réseau de Jordan : bouclage de la sortie sur l'entrée (voir figure ci-dessus).
Réseau d'Elman : bouclage des sorties d'une couche sur la précédente.
Réseau à boucle locale : bouclage de la sortie d'un neurone sur lui-même.
Réseau totalement récurrent : bouclage des sorties de chaque neurone sur tous les autres neurones.

Ce type de réseaux, ne **rentrant pas** dans le **cadre** classique de la **prévision statistique** (puisque manipulant des sorties plutôt que des erreurs) ne sont **pas très bien adaptés à la prévision**.

Le modèle connexionniste NARMA (2)

Principe : Le modèle NARMA procède de la **même logique** que celle qui du modèle AR nous a fait développer le modèle NAR. Il s'agit donc de **remplacer** la **fonction de transfert** du modèle **ARMA** par un **Perceptron Multi-Couches**.



Le modèle connexionniste NARMA (3)

D E F I N I T I O N

Un **processus NARMA(p,q)** général s'exprime sous la forme :

$$x(t) = f[\alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_p \cdot x(t-p) + \beta_1 \cdot e(t-1) + \ldots + \beta_q \cdot e(t-q) + e(t)\ ]$$

où $e(t)$ est un bruit gaussien et f la fonction de transfert d'un Perceptron Multi-Couches.

On en déduit le **prédicteur NARMA(p,q)** associé :

$$\hat{x}(t) = f[\alpha_0 + \alpha_1 \cdot x(t-1) + \ldots + \alpha_p \cdot x(t-p) + \beta_1 \cdot e(t-1) + \ldots + \beta_q \cdot e(t-q)\ ]$$

Propriétés :

Les processus NARMA sont **stationnaires**.

Les processus ARMA étant une sous-classe des processus NARMA, le **théorème d'optimalité** est aussi **valable**.

Les processus NARMA peuvent être **non-linéaires**.

Applications :

Virtuellement **toutes** les séries **stationnaires** (pratiquement, il est inutile d'employer des modèles aussi complexes pour des séries correctement prévues par des modèles ARMA).

Inconvénients:

Nécessité de la **stationnarité** (au moins au second ordre).

Disposant d'un **nombre fini** de point d'apprentissage, le **théorème d'optimalité** est **sans objet**.

Coût en terme de **temps de calcul**.

Difficulté à sélectionner les **dimensions** adéquates.

Le modèle statistique ARIMA

D E F I N I T I O N

Un **processus ARIMA(*p*,*q*,*d*)** est un processus dont la dérivée (au sens de la différenciation) d'ordre *d* est un processus ARMA(*p*,*q*).

Le **prédicteur ARIMA(*p*,*q*,*d*)** associé est donc l'intégrale d'ordre *d* du prédicteur associé au processus ARMA(*p*,*q*).

Nous avons vu précédemment que pour la majorité des séries temporelles, il existait un ordre de différenciation *d* tel que la série résultant de cette différenciation sous **stationnaire**. Ces séries peuvent donc être approchée avec une précision arbitraire par un processus ARMA bien choisi. La **classe** des processus **ARIMA** est donc extrêmement **large**. C'est en fait la classe la plus usitée par les prévisionnistes.

La **différenciation** permettant le passage d'un processus ARIMA(*p*,*q*,*d*) *x* à un processus ARMA(*p*,*q*) *z* peut s'écrire :

$$x^0(t) = x(t)$$

$$n > 1 \Rightarrow x^n(t) = x^{n-1}(t) - x^{n-1}(t-1)$$

$$z(t) = x^d(t) = \sum_{i=0}^{d} (-1)^i \cdot \mathrm{C}_i^d \cdot x(t-i)$$

Une **généralisation** simple consiste à choisir plus simplement les termes de la décomposition dans **R** plutôt que dans **Z** :

$$z(t) = \sum_{i=0}^{d} \alpha_i \cdot x(t-i)$$

Nous **reconnaissons** là un processus AR. La transformation de *x* vers *z* peut donc être généralisée comme une modélisation AR.

Le modèle statistique ARARMA

D E F I N I T I O N

Un **processus ARARMA(*p*,*q*,*d*)** est un processus dont l'application d'une transformation AR(d) fait un processus ARMA(p,q).

Le **prédicteur ARARMA(*p*,*q*,*d*)** associé est donc le prédicteur résultant de l'application de la transformation inverse au prédicteur associé au processus ARMA(p,q).

Propriétés : Les processus ARIMA et ARARMA peuvent être **non-stationnaires**.

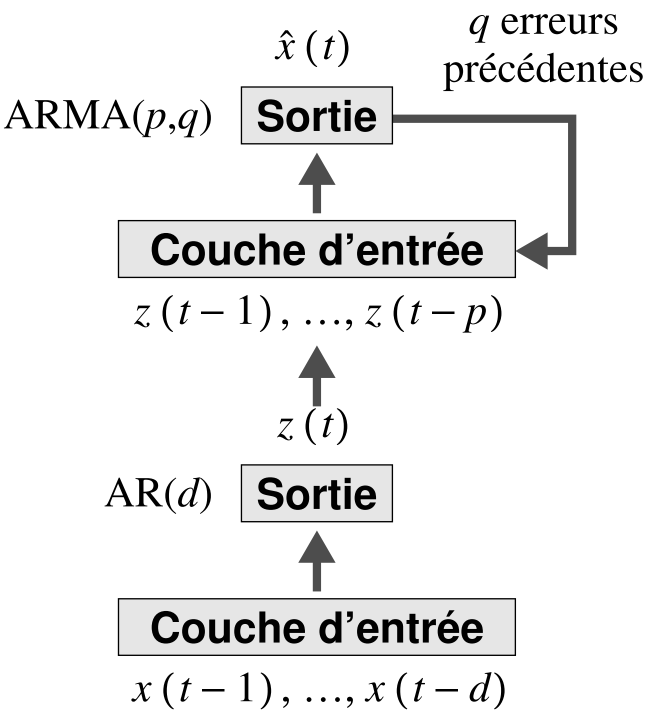
Ils sont une **sur-classe** de tous les **processus statistiques** étudiés jusqu'à présent (et de plus, les processus ARARMA sont une sur-classe des processus ARIMA).

Applications : Virtuellement **toutes** les séries **stationnaires** et un **nombre important** de séries **non-stationnaires** (séries dont une des dérivée est stationnaire dans le cas ARIMA et dont une des transformations AR est stationnaire pour les modèles ARARMA).

Inconvénients: **Linéarité**.

Dans le cas des processus ARARMA, il est **difficile** de trouver la **transformation AR optimale**.

Le modèle connexionniste δ-NARMA (1)



Un modèle ARARMA(p,q,d)

D E F I N I T I O N

Neurone ε-NARMA

Par **analogie** avec les passages statistique - connexionnisme pratiqués jusqu'à présent un **modèle connexionniste** généralisant les **modèles ARARMA** (et par la-même les modèles ARIMA) devrait être basé sur un neurone ayant les **propriétés** suivantes :

- **Stockage** des valeurs **précédentes** de sa **sortie** permettant de les propager à la couche suivante.
- **Bouclage** de ses ***q* erreurs** précédentes sur lui-même.

Le modèle connexionniste δ-NARMA (2)

D E F I N I T I O N

Réseau δ-NARMA

Nous nommerons réseau δ-NARMA un Perceptron Multi-couches dont les neurones sont des modèles ε-NARMA.

Problème : Comment calculer **l'erreur** en sortie d'un **neurone** qui n'**appartient pas** à la **couche de sortie** ?

Solution : La **rétropropagation du gradient** peut être vu comme étant une **rétropropagation de l'erreur**. De l'**erreur sur la couche de sortie**, nous pouvons calculer une **erreur virtuelle** sur la **couche précédente** et ainsi de suite. Nous pouvons donc définir une erreur sur chacun des neurones d'un réseau utilisant cet algorithme ou un de ses dérivés (Back-Propagation Through Time par exemple).

L'**extrapolation** la plus simple du modèle ARARMA dans le formalisme **connexionniste** est donc un réseau δ-NARMA pourvu de trois couches :

Une couche d'entrée.
Une couche cachée pourvue d'un neurone ε-NARMA.
Une couche de sortie pourvue d'un neurone ε-NARMA.

Mais un réseau δ-NARMA peut être beaucoup **plus complexe** en **augmentant** le nombre de couche ou de **neurone** par couche.

Plus de couches → **Augmente la capacité de calcul**

Plus de neurones → **Angmente la segmentation du signal**

Le modèle connexionniste δ-NARMA (3)

Propriétés : Les processus δ-NARMA sont une **sur-classe** des processus **ARIMA** et **ARARMA**.

Les processus δ-NARMA peuvent être **non-stationnaires**.

Les processus δ-NARMA sont **non-linéaires**.

Applications : Théoriquement **toutes les séries** stationnaires ou non, linéaires ou non, à partir du moment où elles peuvent être rendu stationnaire par le passage dans un réseau δ-NARMA.

Inconvénients: L'algorithme d'apprentissage, s'il est simple, est cependant **coûteux** en terme de **temps de calcul**.

L'**ajustement des paramètres architecturaux** est **délicat** car ils sont très nombreux.

Les réseaux δ-NARMA sont relativement **instable** : un phénomène d'**oscillation** apparaît souvent lors de l'apprentissage.

Conclusion

Nous avons pu étudier les modèles statistiques récurrents les plus usités (ARMA, ARIMA et ARARMA) et décrire leur équivalent dans le formalisme connexionniste (NARMA et δ-NARMA).

Avantage :

Intégrant de façon plus **pertinente** le **temps** dans leur traitement, ces modèles sont beaucoup **plus efficaces** que les modèles **auto-régressifs** dont ils sont une **sur-classe**.

Inconvénients :

Lors de l'apprentissage, un phénomène d'**oscillation** survient (ceci est vrai pour tous les modèles à l'exception du modèle NARMA).

Ces modèles impliquent l'**ajoût** de **paramètres** supplémentaires liés au bouclage des erreurs.

Performances :

Nous avons ici utiliser le **même protocole** de comparaison que dans le cas **auto-régressif**. Nous indiquons pour mémoire les résultats obtenus avec les modèles auto-régressifs.

AR:	1.0
NAR:	0.97
TAR:	0.94
NTAR:	0.97

NARMA:	0.86
ARIMA:	0.90
ARARMA:	0.81
δ-NARMA:	0.69

Le meilleur modèle pour la prévision telle que nous l'avons considérée est donc le modèle connexionniste **δ-NARMA**. C'est aussi celui qui a demandé le **plus de développement** quant à son **architecture**.

Hybridation

Comprendre

Choisir une manière de représenter les variables symboliques.

Comprendre quelles sont les variables pertinentes.

Prédire

Intégrer les variables symboliques aux modèles univariés.

Evaluer

Evaluer la capacité des modèles univariés à s'affranchir des variables symboliques.

Choisir une manière de représenter les variables symboliques

Variable symbolique
- Pas de topologie (pas de voisinage).
- Des valeurs dans un ensemble fini.

E X E M P L E

Les données calendaires

Pas de topologie : on ne peut introduire une topologie entre la nature des différents jours (ouvré, férié ou de vacances A,B et C).

Des valeurs dans un ensemble fini : un jour est un jour ouvré, un jour férié ou un jour de vacance zone A, B ou C.

Représentation :

Pour s'assurer de l'**abscence de topologie**, nous utiliserons des variables **binaires**.

Il nous faut donc décomposer les **symboles** en variables **binaires** (classes). Nous définirons donc 4 classes différentes :

La classe des jours fériés (y compris les week-ends).
Les vacances de la zone A.
Les vacances de la zone B.
Les vacances de la zone C.

Cette représentation nécessite de définir **une classe** par **effet observé**. En particulier, si nous introduisons un nouvel évenement (par exemple les 24 heures du Mans), il nous faudra définir une nouvelle classe qui définirait l'occurence des 24 du Mans.

Comprendre quelles sont les variables pertinentes

D E F I N I T I O N

Statistique de Fischer

La statistique de Fischer *F* permet de définir la probabilité de **rejet** de l'**hypothèse nulle** entre un signal $x(t)$ et l'appartenance du jour t à une classe donnée (probabilité que les deux signaux soient dépendants) :

> La statistique de Fischer *F* relative au signal $x(t)$ et à un ensemble de classes donné est le rapport de la **variance entre chaque classe** sur la somme des variances à l'**intérieur de chaque classe**.

Nous pouvons ainsi dire s'il existe un lien entre le fait qu'un jour t soit férié et le signal observé $x(t)$.
Cependant, nous savons que le fait qu'un jour soit férié peut aussi influer sur les jours suivants et précédents. Nous introduisons donc la statistique de Fischer glissée.

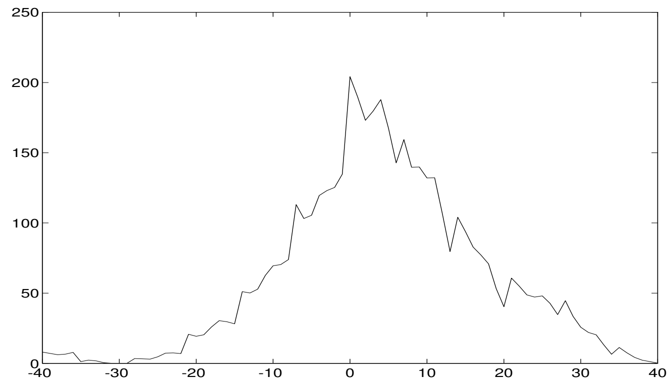
D E F I N I T I O N

Statistique de Fischer glissée

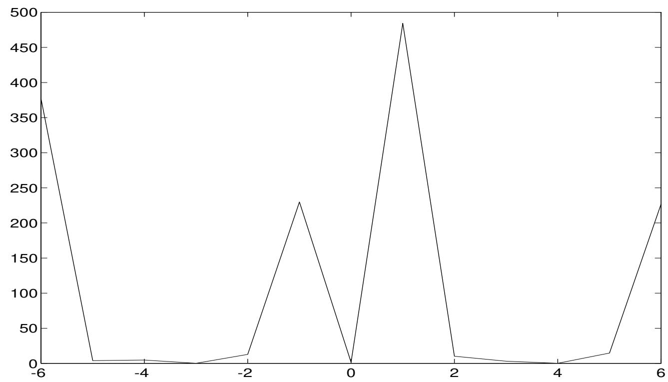
La statistique de Fischer glissée $F(d)$ permet de définir la probabilité de **rejet** de l'**hypothèse nulle** entre un signal $x(t+d)$ et l'appartenance du jour t à un classe donnée (probabilité que les deux signaux soient indépendants).

E X E M P L E

Fischer glissé des variables calendaires sur le trafic journalier



Vacances zone A



Jours fériés

Evaluer la capacité des modèles univariés à s’affranchir des variables symboliques

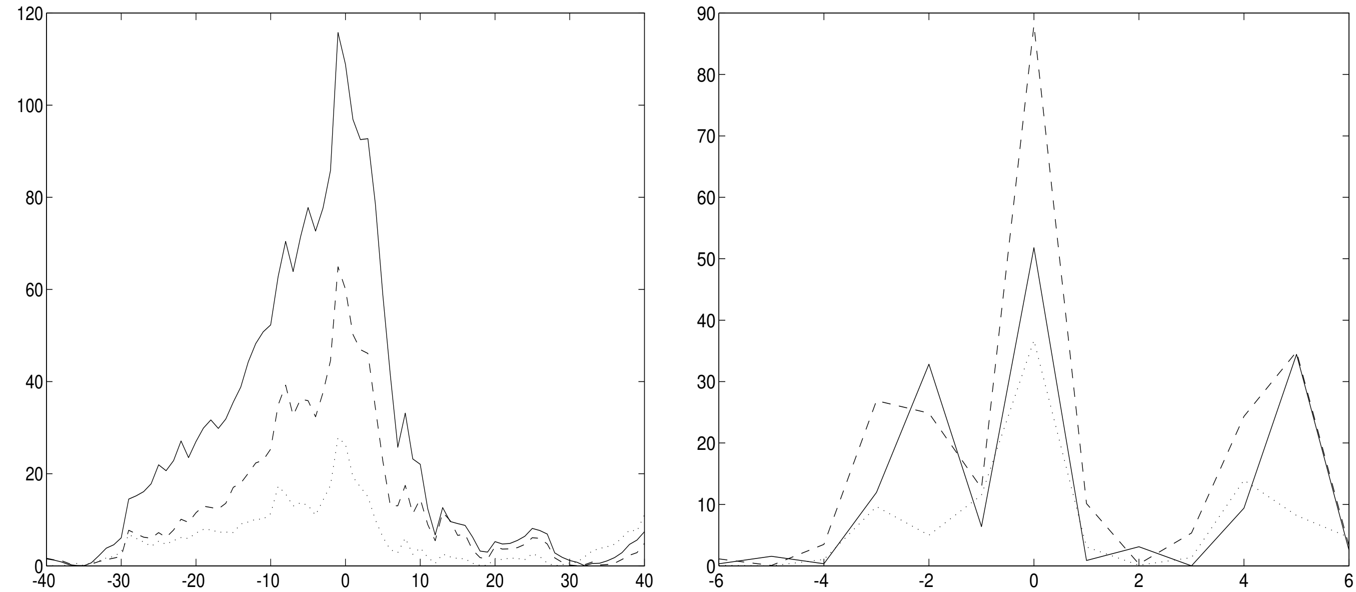
Connaître la **corrélation** existant entre le **signal d’erreur** d’un prédicteur univarié (sans variable explicative) et des **variables symboliques** dont on sait qu’elles sont **pertinentes** est un bon moyen d’**évaluer les performances** d’un modèle.

↓

On peut utiliser la **statistique de Fischer glissée** des variables symboliques calendaires pertinentes **sur l’erreur** commise par divers modèles **univariés** à cette fin.

E X E M P L E

Fischer glissé des variables calendaires sur le signal d’erreur de divers prédicteurs univariés dans le cas du trafic journalier



Vacances zone A **Jours fériés**

Plein : modèle NAR Tirets : modèle NARMA Points : modèle d-NARMA

Cette évaluation confirme les bons résultats des modèles δ-NARMA.

Intégrer les variables symboliques aux modèles univariés (1)

INTEGRATION PAR FUSION

Considérer les variables **symboliques** comme des variables **classiques**.

Déterminer à l'aide de la **statistique de Fischer glissée** la les délais (positif k et négatif $-m$) à partir desquels il n'y a **plus de corrélation**.

Pour prédire $x(t)$, faire intervenir non seulement les **variables** relatives au jour t mais aussi celles relatives aux **jours** $\boldsymbol{t-m,...,t+k}$.

C'est la méthode **la plus efficace** pour les **modèles connexionnistes** car étant d'une complexité ajustable, ils sont suffisamment **puissants** pour intégrer correctement cette information. Les modèles de ce type recevront un X (comme eXogène) à la fin de leur acronyme (NARX,...).

Au contraire, les **modèles statistiques** ne sont **pas assez complexes** pour intégrer cette information **sans nuire** à la partie **univariée** de la prévision.

Intégrer les variables symboliques aux modèles univariés (2)

INTEGRATION PAR SEGMENTATION

Déterminer les ***n*** **configurations des variables explicatives** provoquant des comportements **spécifiques**.

Considérer un ensemble de ***n*** **prédicteurs auto-régressifs**.

Choisir parmi les *n* prédicteurs celui qui sera utilisé dans pour chaque point de la série en fonction de la **configuration** des **variables explicatives**.

Cette méthodes est la plus **performante** pour les **modèles statistiques**. Elle soulève cependant un certain nombre d'**objections** :

Les n **prédicteurs** étant **indépendants**, il est impossible de tirer parti de l'erreur commise par l'un d'eux sur un point atypique pour **corriger** la prévision du point suivant.

Il est très **délicat** de trouver les **configurations** nécessitant un **prédicteur spécifique**.

Nous avons tenté d'utiliser un **modèle NTAR** dont la **porte** était purement **symbolique** mais les résultats de l'apprentissage étaient peu satisfaisants (équilibrage très complexe à effectuer entre les experts et les portes).

Conclusion

Les **variables symboliques** (calendaires par exemple) ont une **influence très importante** sur la **qualité de la prévision**. Cependant, aussi bien du point de vue connexionniste que du point de vue statistique, l'hybridation est le parent pauvre de la recherche. C'est sans doute pourquoi nous pouvons observer aujourd'hui un regain d'intérêt pour les systèmes hybrides symbolique/connexionniste.

La **méthodologie** que nous proposons, basée sur la **statistique de Fischer glissée**, est une stratégie nouvelle mais modeste visant à fournir des **outils automatisés** de manipulation et de selection de **variables symboliques**.
Elles constituent par la même le pendant symbolique aux méthodes classiques du **traitement du signal** (spectre de densité de puissance, inter et auto corrélations) utilisées dans le cas de variables explicatives numériques.

D'un point de vue plus général, les résultats obtenus semblent indiquer que les **modèles connexionnistes**, malgré leur traitement très fortement topologique, sont capable sous certaines conditions de **manipuler efficacement des données symboliques**.

Performances :

Nous n'avons pu tester notre approche que **sur la série journalière**, les composantes explicatives symboliques des deux autres séries étant peu significatives. De plus, nous nous sommes cantonés au cas de l'**hybridation par fusion**.

ARX:	0.95
NARX:	0.22
TARX:	0.78
NTARX:	0.89

NARMAX:	0.19
ARIMAX:	0.82
ARARMAX:	0.75
δ-NARMAX:	0.17

La **différence de performances** entre les modèles **statistiques** et **connexionnistes** est ici **notable**. De plus, nous pouvons observer le **nivellement des performances** des divers modèles **connexionnistes**. Ceci est du en grande partie au fait que l'intégration des variables calendaires aux modèles rend le **traitement temporel** beaucoup plus **simple**.

Synthèse

Nos travaux :

Approfondissement des **comparaisons** entre modèles **statistiques** et **connexionnistes** (notamment TAR/NTAR et ARARMA/δ-NARMA).

Développement des **modèles δ-NARMA**.

Utilisation de la **statistique de Fischer glissée** à des fin d'évaluation et d'ajustement architecturaux.

Etude de **modèles spécifiques** pour la fusion s**ymbolique/connexioniste**.

Implémentation d'un **outil de simulation** permettant d'utiliser tous les modèles vus plus haut.

Les **références bibliographiques** relatives à cet exposé sont disponibles dans le rapport technique ENST :

D. Bonnet, **V. Perrault** et **A. Grumbach** (1996), «δ-NARMA Neural Networks: A New Approach to Signal Prediction», rapport technique ENST 96C008, Ecole Nationale Supérieure des Télécommunications, Paris.