UCL
Université
catholique
de Louvain

Faculté
des
Sciences Appliquées

ANALYSE ET PRÉDICTION DE DÉBIT DE RIVIÈRES PAR DES MÉTHODES NON LINÉAIRES

ERIK QUAEGHEBEUR

Promoteur :
Vincent Wertz

Assistance :
Amaury Lendasse & Luc Moens

MÉMOIRE présenté en vue de l'obtention
du diplôme d'études approfondies en sciences appliquées

Année académique 2001–2002

Survol

MÉMOIRE présenté en vue de l'obtention du diplôme d'études approfondies en sciences appliquées

Titre :
Analyse et prédiction de débit de rivières par des méthodes non linéaires
Auteur : *Ir. Erik Quaeghebeur*
Promoteur : *Pr. Vincent Wertz*
Assistance : *Ir. Amaury Lendasse et Ir. Luc Moens*

Unité AUTO du département d'ingénierie mathématique
Faculté des Sciences Appliquées
Université catholique de Louvain
Année académique 2001–2002

Mots clé

prédiction - méthodes non linéaires - réseaux de neurones artificiels - cartes de Kohonen - réseaux à fonctions radiales de base - hydrologie - débit de rivières

Abstract

This text treats of the problem of predicting the flow of a river using past flow measurements, rainfall measurements and rainfall predictions. The objective is to generalize a forecasting method already used to predict daily consumption of electricity. The developed method will be compared to Hydromax, an existing riverflow forecasting model.

The developed method simultaneously generates a series of consecutive flow predictions called a flow curve. This is done by combining separate forecasts for the mean, standard deviation and the normalized profile of the flow curve. Therefore, three separate forecasting models will be used, one for each of the aforementioned flow curve components.

To understand the difficulties involved in riverflow forecasting we first take a look at the relevant hydrological concepts and the data used in constructing and testing the forecasting method. We will principally be interested in the prediction of floods, as these phenomena can have dire socioeconomic consequences if they take place without a timely warning.

Some of the forecasting models will be based on mathematical techniques derived from the field of artificial neural networks. As an introduction we will shortly elaborate on this field before presenting a more profound study of the two derived techniques we will use. These are the so-called 'self-organizing maps' and 'radial basis function networks'.

An essential part of the forecasting method are linear and nonlinear regression models. We shall take a look at these parameterized prediction models, the associated prediction errors and the parameter estimation involved in constructing them.

Thus being well prepared, we will have at this stage a close look at the Hydromax model and explain the forecasting method we have developed. They will be compared in terms of the data needed and the hydrological knowledge involved.

Finally, we will show how the developed method is used in practice. The obtained results will be compared with those obtained by Hydromax, and remarks will be made about possible improvements.

We will conclude with showing that the developed method holds promise, but is not yet suitable for practical applications. Suggestions for further research are also included.

Préface

Ce mémoire est fait dans le cadre de mes études pour un diplôme d'études approfondies (DEA) en sciences appliquées, spécialisation mathématiques appliquées. Mon intérêt pour la mathématique s'est accru pendant la dernière année de mes études d'ingénieur en physique à Gand et j'ai commencé ce DEA avec en tête un but clair. Je voulais d'abord découvrir les différents aspects des mathématiques appliquées. En plus, cette formation devait me préparer à la recherche dans un des domaines des mathématiques appliquées.

Comme le DEA dans sa totalité, le travail que j'ai fait pour ce mémoire comprenait une étude et une application de concepts de divers domaines scientifiques. Les plus importantes étant la théorie des réseaux de neurones, l'hydrologie et l'identification de systèmes.

J'ai écrit ce texte pour un public relativement étendu en tête. Un public ne pas seulement composé d'experts dans les domaines touchés dans ce texte, mais aussi comprenant des non-experts. Il y a bien sûr comme prérequis une connaissance de base des mathématiques.

Pour faire des simulations, j'avais besoin de mesures de débits et de mesures pluviométriques. Les données horaires de débit et les données pluviométriques horaires utilisées sont mises à disposition gracieusement par le Ministère wallon de l'Équipement et des Transports (MET), Voies Hydrauliques (DG.2–D.212), Service d'Études hydrologiques (SETHY).

Je voudrais remercier A. Lendasse et L. Moens pour l'aide qu'ils m'ont donnée dans le cadre de ce mémoire. Aussi J.A. Lee mérite des remerciements pour avoir mis à disposition son logiciel pour les réseaux à RBF.

Finalement, je veux encore dire que c'est grâce au soutient de mes parents que j'ai pu faire ce DEA et que c'est grâce au suivi du professeur Wertz que cette formation s'est déroulée sans encombre.

Erik Quaeghebeur
12 juin 2002

Table des matières

Table des figures

Chapitre 1

Introduction

1.1 Le but et la justification de ce travail

Le but de ce mémoire est d'*étudier une méthode pour prédire le débit de rivières*, un phénomène naturel qui dépend de manière non-triviale des conditions météorologiques et géographiques. Ici, prédire c'est utiliser de l'information, disponible à l'instant présent, pour calculer une estimation d'une certaine grandeur pour un ou plusieurs instants dans le futur. Pourquoi est-il utile d'avoir des estimations du débit pour des instants futurs d'une rivière à un endroit donné ? Pour répondre à cette question, il ne faut que penser à deux des désastres les plus catastrophiques dans le monde : les inondations et les périodes de sécheresse.

Des grandes parties du monde dépendent des rivières pour les nourrir, c'est l'eau de ces rivières qui est utilisée pour irriguer les champs. De meilleures estimations de l'évolution du débit peuvent permettre, par régulation, de mieux utiliser l'eau disponible.

Ce n'est pas seulement le manque d'eau qui peut être désastreux, les grands accroissements inattendus[1] des crues de rivières entraînent aussi des problèmes importants. On voudrait évidemment pouvoir éviter le danger immédiat pour les gens de noyer. En plus, il y a aussi le problème de villages et de fermes isolés et les ennuis qui s'en suivent pour la population. Si ici en Europe ces problèmes restent relativement limités, il y a aussi, ce qui est particulièrement important dans nos régions, les conséquences économiques. Pour le secteur agricole il y a la perte des bestiaux et de récolte. Pour tout les secteurs il y a des problèmes de transport et des dégâts causés par l'eau. La figure 1.1 est une illustration des effets typiques d'une inondation dans nos régions.

1. Ici, il faut contraster les inondations inattendues avec les inondations structurelles, comme en Bangladesh pendant le Mousson. Bien que les inondations structurelles peuvent aussi être désastreuses, nous n'allons pas les considérer.

FIGURE 1.1: Photo d'une rue inondée.

1.2 Prédiction de séries temporelles

Dans la section précédente, on a vu ce qu'on veut faire et l'utilité de cette entreprise pour la société. Pour arriver au but il est nécessaire de formaliser la problème. Nous allons la formuler dans le cadre de la *prédiction de séries temporelles*.

Pour notre problème, la série temporelle est une série de débits ou une série de vecteurs de débits ordonnées dans le temps, où l'instant correspondant à l'élément le plus récent est considéré d'être le présent. Le but est de prédire une ou plusieurs éléments futurs de la série. Afin d'y arriver, il faut essayer d'utiliser au maximum l'information pertinente contenue dans la série temporelle même, mais aussi l'information sur l'influence éventuelle d'autres séries temporelles (comme, pour nous, celle contenant les données de précipitation). Des différentes façons d'utiliser ces informations donnent lieu au différents modèles de prédiction.

Les modèles de prédiction peuvent être caractérisé par le degré de connaissance théorique utilisé concernant le phénomène considéré (pour nous, les variations des débits). Nous examinerons un modèle qui est construit partiellement sur des connaissances théoriques et un modèle qui utilise (presque) aucune connaissance théorique.

Les modèles qui ne sont pas construits totalement sur base des connaissances théoriques, sont construits à partir d'un *ensemble d'apprentissage* constitué d'observations venant du système. Cet ensemble est ainsi appelé parce qu'il contient des situations que le modèle peut «apprendre» à prédire. Si, après cette phase d'apprentissage, le modèle de prédiction est aussi capable de prédire des situations ne pas présent dans l'ensemble d'apprentissage, on parle d'apprentissage avec généralisation.

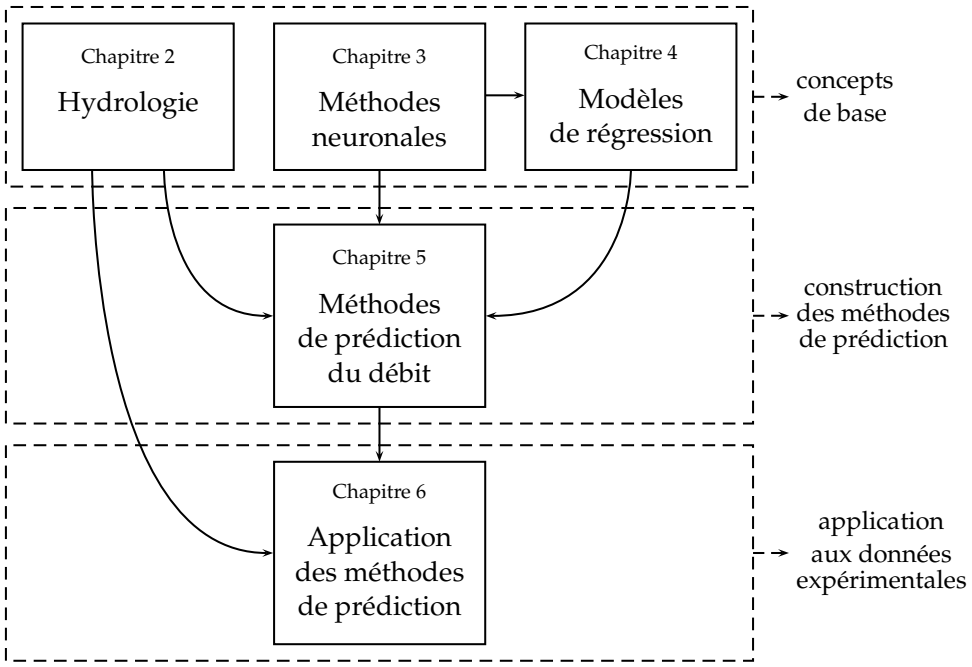

FIGURE 1.2: La structure de ce mémoire.

1.3 Le contenu et la structure du mémoire

Hors cette introduction et la conclusion, ce mémoire contient cinq chapitres. Les titres de ces chapitres et leurs liens sont montrés dans la figure 1.2.

La première et la plus grande partie de ce mémoire consiste de trois chapitres décrivant les concepts et les méthodes qui forment la base des méthodes de prédiction.

Le chapitre 2 traite d'abord les concepts de l'*hydrologie* dont on a besoin pour pouvoir comprendre la problématique de la prédiction de débit de rivières. Une fois que ces concepts sont clair, on peut présenter les données dont nous disposons. Ces données sont des mesures faits dans deux bassins versants de la Wallonie. Ceci entraîne que nous n'allons jamais avoir à faire avec des situations de sécheresse et nous nous intéressons surtout à pouvoir prédire des crues dangereuses et éventuellement nuisibles.

Le chapitre 3 traite les *méthodes neuronales*. Deux de ces méthodes vont jouer un rôle central dans la méthode de prédiction qu'on proposera (c'est à dire, les *cartes auto-organisatrices* et les *réseaux à fonctions radiales de base*). Comme ces deux méthodes ne sont pas omniprésent dans le domaine de la prédiction des séries temporelles, on les traite de façon détaillé et on les introduit en décrivant ses origines, les réseaux de neurones artificiels.

Dans le chapitre 4, on examinera des *modèles de régression* couramment utilisé pour la prédiction de séries temporelles. On commencera en examinant les modèles paramétriques en général, comment de déterminer les paramètres

y associés et comment de quantifier la qualité d'un modèle construit. Dans une deuxième étape on examinera deux modèles pratiques, un linéaire (ARX) et un non linéaire (sur base d'un réseau à fonctions radiales de base).

Comme ces trois chapitres décrit ci-dessus donnent des notions fondamentales, il est clair que les lecteurs, qui sont déjà familiers avec les concepts décrits, peuvent les lire plus vite. Afin de garantir une lecture aisée, on a essayé d'assez souvent clarifier les notations utilisées.

La deuxième partie consiste du chapitre 5, qui décrit deux méthodes de prédiction. Il est évident qu'il y a déjà beaucoup de travail et de recherche faits sur la prédiction de débits. Un des résultats obtenus est le modèle Hydromax (voir [1]). Le logiciel[2] basé sur ce modèle est à présent utilisé par le Service d'Études hydrologiques de la Wallonie (SETHY). Nous allons examiner ce modèle dans la première partie de ce chapitre (section 5.2). Ceci nous permettra de comparer la méthode qu'on développera dans ce mémoire avec une méthode de prédiction déjà existante. On fera une comparaison des résultats, mais aussi des connaissances requises pour l'application dans la pratique. La deuxième partie (section 5.3) sera notamment consacrée à la méthode que nous avons développée. Cette méthode est basée sur une méthode déjà utilisée dans d'autres situations, notamment la prédiction de consommation d'électricité (regardez par exemple [2]). Elle a comme but de prédire, non pas une seule valeur future par étape, mais une série de valeurs futures. On attaque ce problème en le divisant en trois parties : la prédiction indépendante de la moyenne de la série, l'écart type de la série et une version normalisée de la série appelée le profil. Les problèmes liés à la prédiction de la moyenne et de l'écart type sont résolus en utilisant des méthodes de régression linéaires et non linéaires. Pour la prédiction du profil on va utiliser des cartes auto-organisatrices[3] (*le* livre de référence est [3]).

Quand dans la deuxième partie on examine les modèles de prédiction d'un point de vue plutôt théorique, la troisième partie (chapitre 6) consiste d'une application de ces méthodes aux données décrites dans §2.6. Les modèles de régression pour la prédiction de la moyenne et de l'écart type sont spécifiés, et les résultats de ces modèles sont examiné. Pour la prédiction du profil, nous éclaircissons la méthode en donnant des exemples pratiques. Les résultats finals de notre méthode sont comparées avec ceux obtenus avec Hydromax. Les points pratiques de ces deux méthodes de prédiction y sont aussi examinés. Ce chapitre se finit avec quelques suggestions pratiques pour améliorer la méthode qu'on a développée.

A la fin de chaque chapitre, nous donnerons, dans une section appelé «Bibliographie», les sources qu'on à consulté pour nous aider à écrire le chapitre en question.

---

2. Hydromax est le nom du logiciel, mais nous l'utiliserons aussi pour le modèle même.
3. Les cartes auto-organisatrices sont aussi appelé *cartes de Kohonen*, d'après leur inventeur.

En dehors de ces chapitres, on a aussi inclus deux annexes. L'annexe A contient quelques remarques complémentaires utiles, mais qu'on n'a pas incluses dans la texte même pour éviter des passages alourdissants. L'annexe B contient la code des programmes utilisés dans le cadre de ce mémoire.

1.4 Notation

Ci-dessous se trouve une liste d'exemples type de la notation mathématique utilisée. Ces exemples montrent toutes les caractéristiques des symboles utilisées pour désigner un concept : majuscule ou minuscule, alphabet romain ou grecque et quelle 'région' de ces alphabets.

$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	nombres naturels, intègres et réels
$\mathbb{Z}^+, \mathbb{R}_0^+$	nombres intègres strictement positifs, réels positifs
u, x, y, z	variables
$\lambda, \alpha, \beta, a, b$	coefficients et constantes réelles
i, j, k, l	indices intègres
α, β, γ	indices générales
m, n, l	le nombre d'éléments dans un ensemble ou le nombre de composantes d'un vecteur
Q, PB	grandeurs hydrologiques
$\hat{Q}, \hat{y}_k$	estimations
$\boldsymbol{u}, \boldsymbol{\pi}, \boldsymbol{PN}$	vecteurs
V, H, T, Y	ensembles[4]
f, g	fonctions
h, m^3/s	unités des grandeurs

On utilise bien sûr aussi d'autres types de notations,

$\{ \quad \}$	délimiteurs des éléments d'un ensemble
$[\quad]$	délimiteurs de groupement
$(\cdot)$	délimiteurs de l'argument d'une fonction[5]
$(a_1, \ldots, a_n)$	composantes d'un vecteur
$\langle \boldsymbol{a}, \boldsymbol{u} \rangle$	produit scalaire quelconque
$\boldsymbol{a} \cdot \boldsymbol{u}$	produit scalaire classique

1.5 Bibliographie

Pour la section 1.2 on a pu utiliser quelques idées de [4], un article qui donne un aperçu très lisible du domaine de la prédiction de séries temporelles.

4. Confusion avec les grandeurs hydrologiques sera évité par le context.
5. Le point « · » est utilisé quand on ne spécifie pas d'argument.

Chapitre 2

Hydrologie

2.1 Introduction

Dans ce chapitre, on esquissera le contexte hydrologique de ce mémoire. Ceci aidera à bien comprendre la signification des grandeurs utilisées et les influences physiques déterminant celles-ci. On parlera ensuite de la manière dont les données sont acquises et on décrira les origines de ces données. Finalement, nous faisons une analyse initiale des données.

2.2 Hydrologie

L'hydrologie est la science qui a pour objet l'étude des eaux et leurs propriétés.

On ne va pas faire une étude de l'hydrologie ici, mais il est quand-même nécessaire de considérer quelques concepts de base. Il sera aussi utile de parler de quelques caractéristiques des systèmes hydrologiques étudiés ici. Ceci permettra de mieux comprendre les approches utilisées pour la prédiction décrites dans la suite.

2.2.1 Systèmes hydrologiques

La théorie des systèmes peut être utilisée pour construire des modèles pour les phénomènes hydrologiques.

Ce qui nous intéresse, ce sont les modèles entrée-sortie. Les entrées sont des mesures de débit et des mesures ou des prédictions de pluie. La sortie est une estimation de débit. Ces modèles ont des paramètres dépendant du système spécifique étudié et éventuellement des saisons.

2.2.2 Concepts hydrologiques

Un *bassin versant* d'une rivière est une région avec des limites géographiques déterminées par le fait que toutes les eaux (venant de sources, glaciers,

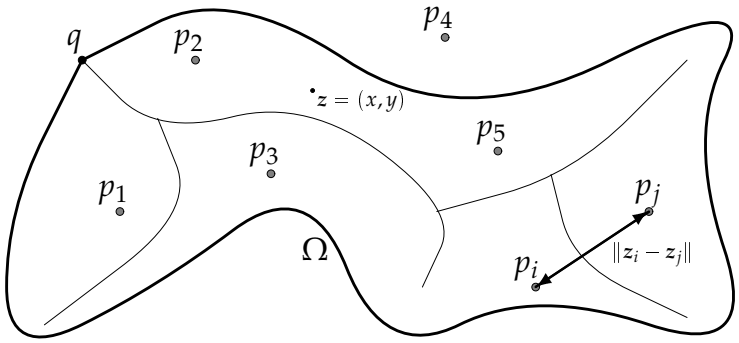

FIGURE 2.1: Bassin versant.

précipitations) dans cette région ne peuvent qu'en sortir (par ruissellement naturel) par la rivière concernée. Il est aussi possible de parler d'un bassin versant d'une rivière à un point spécifique sur la rivière.[1]

Le *débit* d'une rivière à un certain point le long de son cours est le volume d'eau qui passe par ce point par unité de temps.

On rencontrera plusieurs grandeurs qui ont comme nom pluie, notamment la *pluie brute* et la *pluie nette*. Ces grandeurs sont des hauteurs d'eau accumulées par unité de temps.

Une partie de l'eau stockée dans le bassin peut (re)gagner l'atmosphère par *évapotranspiration*. C'est à dire, par évaporation directe ou transpiration de la couverture végétale. L'évapotranspiration a une périodicité journalière (basse pendant la nuit, haute pendant la journée) et une périodicité saisonnière (basse en hiver, haute en été). L'évapotranspiration est mesurée en hauteur d'eau évapotranspirée par unité de temps.

2.3 Mesure des données

Dans la figure 2.1 un bassin versant Ω est schématiquement montré, le bassin est borné par une ligne épaisse, les rivières sont dessinées par des lignes fines. La signification des autres éléments dessinés sera expliquée ci-dessous.

Une mesure de débit est effectuée comme suit. En utilisant un limnigraphe q, situé à un certain point sur la rivière, on mesure la hauteur d'eau en ce point. Avec cette hauteur d'eau on peut calculer le débit à ce point en tenant compte de la forme spécifique du lit de la rivière et de la vitesse de l'eau. Une mesure de débit au temps t est notée $Q(t)$. Le limnigraphe qui nous intéresse est positionné à l'exutoire du bassin. Le débit est exprimé en mètres cubes par seconde, [m^3/s].

1. On peut par exemple parler du bassin de l'Escaut à Gand.

Une mesure de pluie[2] au coordonnées cartésiennes $z = (x, y) \in \mathbb{R}^2$ et au temps t est notée $P(z, t)$. La pluie est mesurée par un certain nombre de pluviomètres p_i localisés en z_i, des points qui ne sont pas nécessairement dans Ω. Les mesures correspondantes sont notées $P_i(t) = P(z_i, t)$. Une mesure de pluie est exprimé en millimètres accumulés pendant une heure, [mm]. Alors $P_i(t)$ est le nombre de millimètres de pluie accumulés entre $t - 1$h et t.

Il y a plusieurs méthodes pour mesurer ou estimer l'évapotranspiration. Nous supposerons comme connu les valeurs caractéristiques saisonnières.[3]

2.4 Précipitation moyenne

La *précipitation moyenne* $\check{P}B(t)$ est la lame d'eau moyenne affectant un bassin versant.[4] Pour calculer $\check{P}B(t)$ sur le bassin versant $\Omega \subset \mathbb{R}^2$ on doit évaluer

$$\check{P}B(t) = \frac{1}{|\Omega|} \int_\Omega P(z, t) dz,$$

avec $|\Omega| = \int_\Omega dz$ la surface du bassin versant.

Vu qu'on n'a que de mesures discrètes, il est nécessaire de faire une estimation de la précipitation moyenne $PB(t)$. Cette grandeur est appelée la *pluie brute*.

Une des méthodes possibles est de prendre la moyenne arithmétique des différentes mesures,

$$PB(t) = \frac{1}{n} \sum_i P_i(t),$$

avec n le nombre de points de mesures. Dans notre méthode de prédiction (§5.3), on utilisera cette estimation.

L'estimation (linéaire, sans biais et de variance minimale) utilisée par Hydromax (voir §5.2) est calculée comme suit,

$$PB(t) = \sum_i \lambda_i P_i(t),$$

avec les coefficients λ_i venant du système de krigeage

$$\begin{aligned} \sum_i \lambda_i \|z_i - z_j\| + \mu &= \frac{1}{|\Omega|} \int_\Omega \|z - z_j\| dz, \quad \forall j, \\ \sum_i \lambda_i &= 1, \end{aligned}$$

avec $\|z_i - z_j\|$ la distance entre z_i et z_j et μ un multiplicateur de Lagrange.

2. Une hauteur de pluie accumulée dans un certain temps donné, ou autrement dit, l'intégration d'une lame d'eau pendant une période de temps.
3. On n'entre pas dans les détails de la mesure de l'évapotranspiration, parce qu'on n'utilisera pas cette grandeur dans notre méthode de prédiction (voir section 5.3), contrairement à Hydromax (voir section 5.2).
4. Cette précipitation moyenne est alors une moyenne prise sur la surface du bassin.

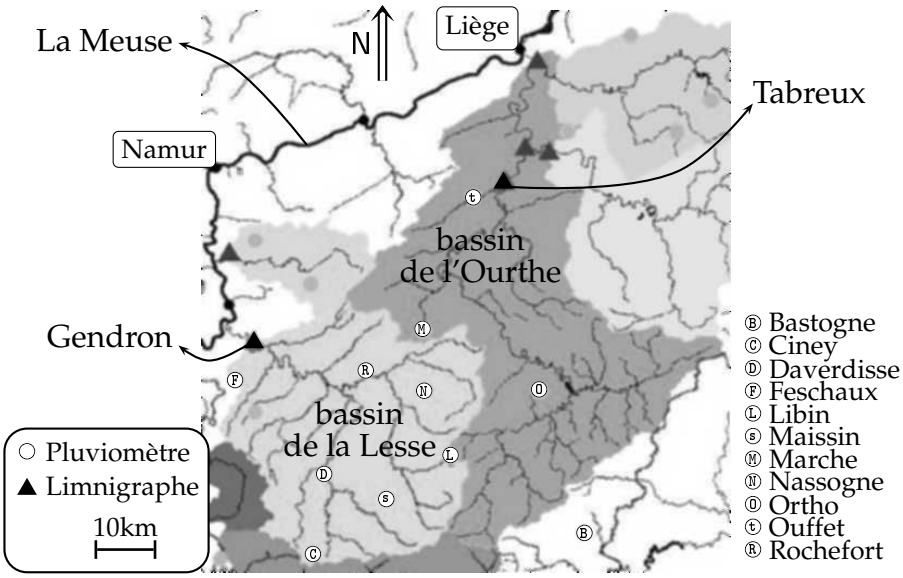

FIGURE 2.2: Carte des bassins versants de la Lesse et de l'Ourthe.

2.5 Données disponibles

Les mesures (les débits et les pluies) à notre disposition sont fournies par le SETHY, le service d'études hydrologiques du ministère wallon de l'équipement et des transports. Le SETHY a un réseau de télémètres dans les différents bassins versants de la Wallonie pour rassembler ces mesures.

Les mesures sont effectuées à des temps de mesure t_k équidistants, c'est à dire $t_k = t_0 + k\Delta t$, avec k entier. Dans notre cas les mesures sont horaires, i.e. $\Delta t = 1\text{h}$ et mesurées à l'heure. Par exemple, t_0 est égale à deux heures du matin le premier janvier 1993.

On dispose donc de mesures de débit $Q(t_k)$ et de mesures pluviométriques $P_i(t_k)$, qui nous permettent de calculer $PB(t_k)$. Les données utilisées[5] pour les prédictions et leurs notations raccourcies sont $Q_k = Q(t_k)$ et $PB_k = PB(t_k)$.

2.6 Présentation des bassins versants

Les deux bassins d'où viennent les mesures sont ceux de la Lesse à Gendron et l'Ourthe à Tabreux. Ces deux rivières sont des affluents du fleuve la Meuse. Elles se trouvent dans la région Sud-Est de la Belgique, plus précisément dans la Famenne et dans les Ardennes.

Dans la figure 2.2 on a donné une carte de la région concernée. Les bassins sont montrés en différents types de gris. En plus, les positions des limnigraphes et des pluviomètres y sont indiquées.

5. Les mesures sont utilisées pour l'estimation des paramètres du modèle de prédiction (aussi appelé apprentissage) et la validation de ce modèle.

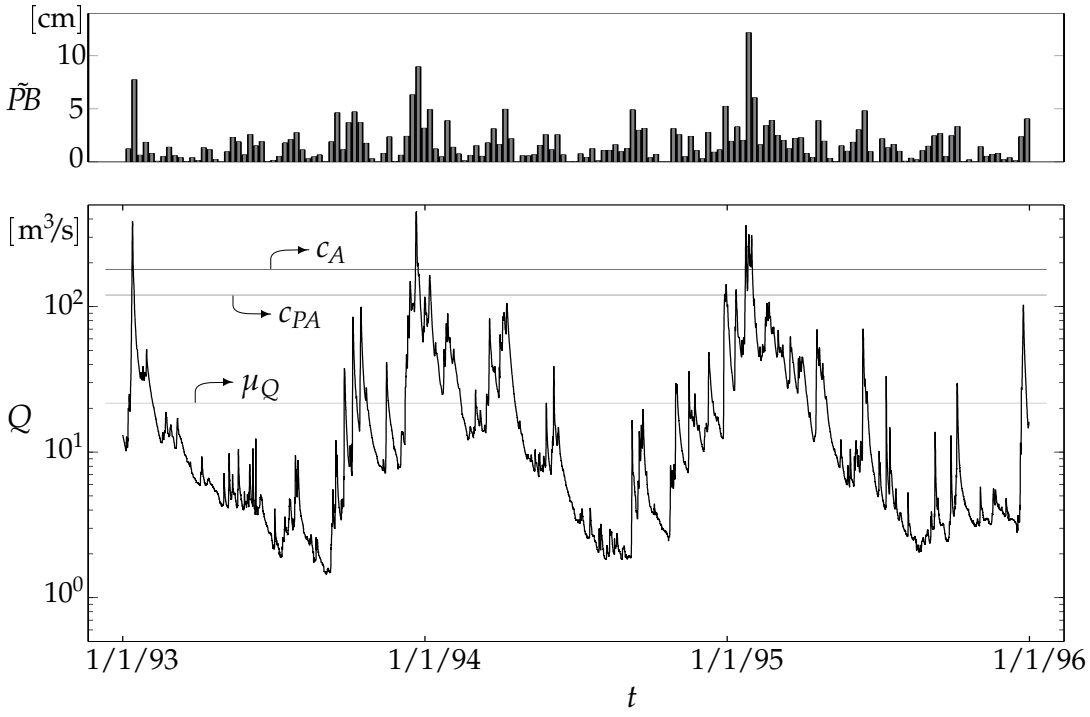

FIGURE 2.3: Mesures de débit à Gendron et la pluie brute (par semaine).

2.6.1 Bassin de la Lesse à Gendron

La première série de données est celle utilisée pour prédire le débit de la Lesse à Gendron.

D'abord on examinera les mesures de débit de la Lesse à Gendron utilisées. Ces mesures couvrent une période de trois années : du 1/1/93 au 31/12/95. Un graphique de l'évolution du débit est montré dans la figure 2.3. Trois niveaux de débit sont indiqués dans la figure.

- Le débit moyen $\mu_Q = 21{,}7\,\mathrm{m}^3/\mathrm{s}$,
- Le seuil de pré-alerte $c_{PA} = 120\,\mathrm{m}^3/\mathrm{s}$,
- Le seuil d'alerte $c_A = 180\,\mathrm{m}^3/\mathrm{s}$.

Les deux seuils sont déterminés par l'expérience. Le débit maximal mesuré dans cette période était $448{,}2\,\mathrm{m}^3/\mathrm{s}$ en décembre '93. Le débit minimal était $1{,}4\,\mathrm{m}^3/\mathrm{s}$ en septembre '93.

Pour le bassin de la Lesse à Gendron on a des mesures de huit pluviomètres, à Ciney, Daverdisse, Feschaux, Libin, Maissin, Marche, Nassogne et Rochefort pour les trois années citées. Si on calcule la pluie brute *PB*,[6] on constate que les valeurs vont jusqu'à 10,6mm (en octobre '93). Dans la figure 2.3 on a aussi ajouté un graphique montrant la pluie brute, non pas accumulée par heure, mais par semaine, d'où la notation $\tilde{PB}$. Cette grandeur est exprimée en centimètres (notation cm).

6. La pluie brute est calculée ici comme moyenne arithmétique des mesures de pluie.

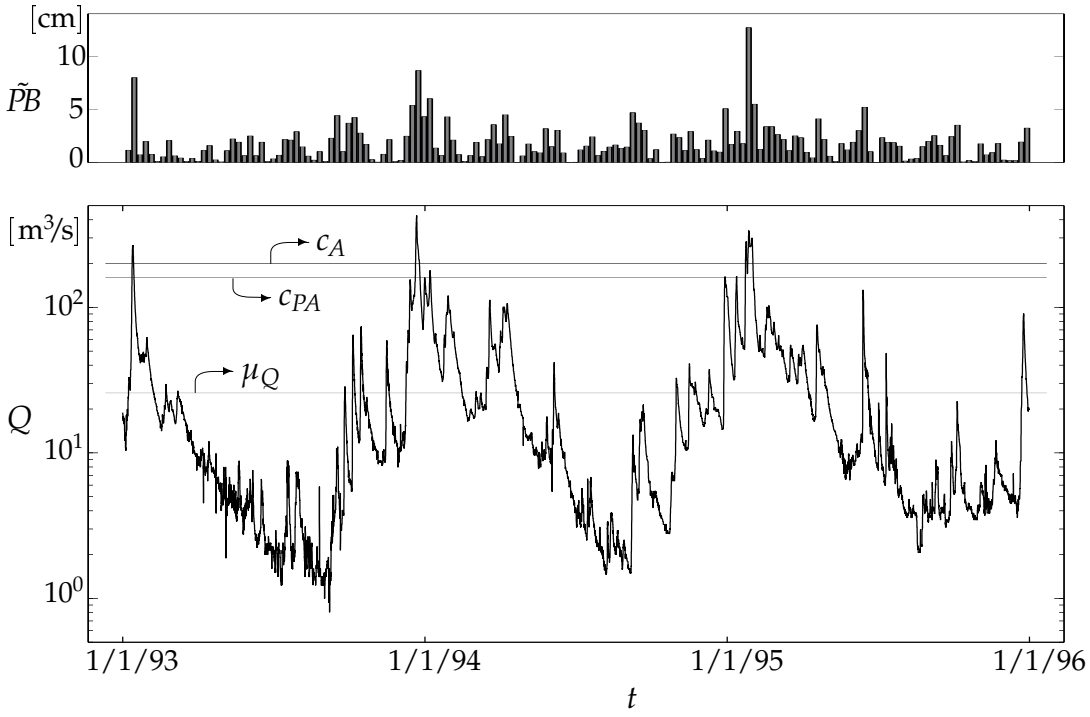

FIGURE 2.4: Mesures de débit à Tabreux et la pluie brute (par semaine).

Remarquez que la façon de presenter les données dans la figure 2.3 exprime le fait d'être une mesure instantanée (débit, courbe continue) ou d'être une mesure intégrée (pluie brute, une hauteur fixe pour chaque semaine).

2.6.2 Bassin de l'Ourthe à Tabreux

La deuxième série de données est celle utilisée pour prédire le débit de l'Ourthe à Tabreux.

Les mesures couvrent la même période qu'avant. Un graphique de l'évolution du débit est montré à la figure 2.4. Les valeurs des trois niveaux de débit spécifique sont :

- Le débit moyen $\mu_Q = 25{,}8\,\mathrm{m^3/s}$,
- Le seuil de pré-alerte $c_{PA} = 160\,\mathrm{m^3/s}$,
- Le seuil d'alerte $c_A = 200\,\mathrm{m^3/s}$.

Le débit maximal était $427{,}7\,\mathrm{m^3/s}$ en décembre '93 et le débit minimal $0{,}8\,\mathrm{m^3/s}$ en septembre '93.

Pour le bassin de l'Ourthe à Tabreux on a des mesures de quatre pluviomètres, à Bastogne, Marche, Ortho et Ouffet. Les valeurs de la pluie brute *PB* vont jusqu'à 9,3mm (en mai '94). Dans la figure 2.4 on a de nouveau un graphique avec la pluie brute par accumulée semaine.

2.6.3 Analyse initiale des données

Si nous regardons les figures 2.3 et 2.4, qui sont très semblables, il y a déjà quelques observations que nous pouvons faire.

La plus importante est le fait que les fortes crues correspondent à des périodes avec beaucoup de pluie. Sachant que la pluie n'est pas nécessairement la seule influence régissant la valeur du débit, on peut quand même déduire des graphiques que la relation entre les deux grandeurs est non linéaire.

Une deuxième observation est la dépendance saisonnière. On a des grands débits pendant l'hiver et des petits débits pendant l'été. Il y a aussi une dépendance analogue pour la pluie, mais elle est beaucoup moins marquée.[7]

Les phénomènes qui nous intéressent beaucoup sont les dépassements des seuils de pré-alerte c_{PA} et d'alerte c_A. Ceux-ci se passent uniquement pendant l'hiver et avec une basse fréquence.

2.7 Bibliographie

La première partie de ce chapitre, §2.2 est basée sur [5, *I, Ch.1*].

Pour §2.3 on a consulté [1] et [5, *I, Ch.3*] et pour §2.4 [1] et [5, *I, Ch.1*].

La section §2.6 est fort influencée par la correspondance avec L. Moens (co-auteur de [1]).

7. Il ne faut pas oublier la différence d'échelle des deux graphiques : logarithmique opposée à linéaire. Le choix d'une échelle logarithmique pour le débit a été fait pour la clarté.

Chapitre 3

Méthodes neuronales

3.1 Introduction

Dans ce chapitre on examinera d'abord les réseaux de neurones. On passera du neurone biologique à travers la modélisation de celui-ci vers les réseaux de neurones artificiels. Ces réseaux sont à l'origine des méthodes mathématiques dites neuronales, dont on étudiera deux exemples : les cartes autoorganisatrices et les réseaux à fonctions radiales de base. Ces deux méthodes neuronales seront utilisées plus tard respectivement pour faire une classification non supervisée et pour faire de la régression non-linéaire.

3.2 Réseaux de neurones

3.2.1 Réseaux de neurones biologiques

Les premiers efforts de recherche sur les réseaux de neurones portent sur le plan biologique. On étudiait les cerveaux et les systèmes nerveux de différents organismes pour comprendre la transmission de signaux dans le corps et pour découvrir les mécanismes de traitement de données utilisées par les animaux.

Il a été découvert que le système nerveux est un réseau constitué d'une interconnexion d'un grand nombres de *neurones biologiques*, un type de cellule spécifique. Cette cellule transmet des signaux et en fait un traitement.

On peut distinguer trois parties distinctes dans le neurone biologique. La première est le *soma*, qui contient le noyau de la cellule. La deuxième partie sont les *dendrites*, les prolongements qui reçoivent les signaux en provenance d'autres cellules. La troisième est l'*axone*, un prolongement unique, qui diffuse le signal émis par le neurone. L'axone est parfois divisé à son extrémité pour pouvoir entrer en contact avec plusieurs autres cellules. Ce contact n'est pas un contact direct entre les membranes du neurone et la membrane de l'autre cellule, mais est assuré par un élément de jonction appelé la *synapse*. Ceci est montré dans la figure 3.1.

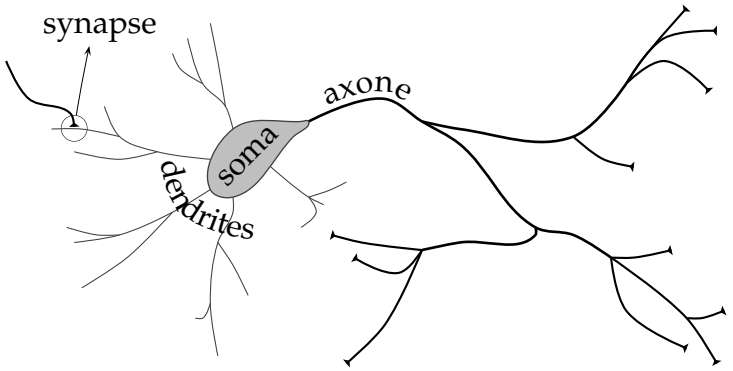

FIGURE 3.1: Le neurone biologique.

La transmission de signaux à travers les chemins de communication est de nature électrique (pour les dendrites et axones) ou chimique (pour la synapse). Les signaux transmis peuvent venir d'autres neurones ou peuvent venir des récepteurs des sens (récepteurs de lumière, pression, température,...) après transformation en signal électrique.

Eventuellement après transformation, les signaux qui se propagent à travers les axones le font sous un forme appelé *potentiel d'action*. Aux synapses, ces signaux sont transformés, une transformation qui dépend des caractéristiques de la connexion synaptique, en *potentiels synaptiques*. Ces potentiels se propagent le long des dendrites vers le soma. Dépendant de l'amplitude du signal intégré arrivant à la transition entre soma et axone, un potentiel d'action y est généré.

Dans la description de la propagation de signaux dans un neurone, on voit qu'il y a deux instances de traitement de signal. La première aux synapses et la seconde à la transition soma-axone. La décision locale prise par le neurone, s'il génère un potentiel d'action ou pas, est à la base du traitement de données global effectué par le système nerveux. Dans le grand réseau de neurones qui est le système nerveux il y a des structures organisées, qui diffèrent selon la nature de l'information recueilli et selon le traitement effectué par cet ensemble de neurones.

3.2.2 Modélisation de réseaux de neurones biologiques

Une étape importante dans le domaine des réseaux de neurones a été prise quand on est passé des observations empiriques à la modélisation. On a fait de la modélisation pour comprendre le fonctionnement du système nerveux à partir d'éléments formels. Ces éléments formels utilisent les propriétés connues des neurones et synapses pour décrire les fonctions remplies par ceux-ci.

Le premier modèle à été proposé par McCulloch et Pitts en 1943 dans [6]. Ce modèle est bien sûr une approximation relativement simple du neurone

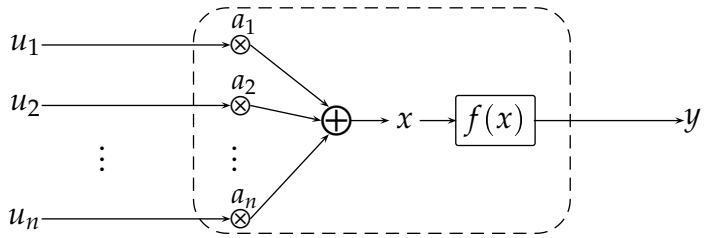

FIGURE 3.2: Le modèle d'un neurone.

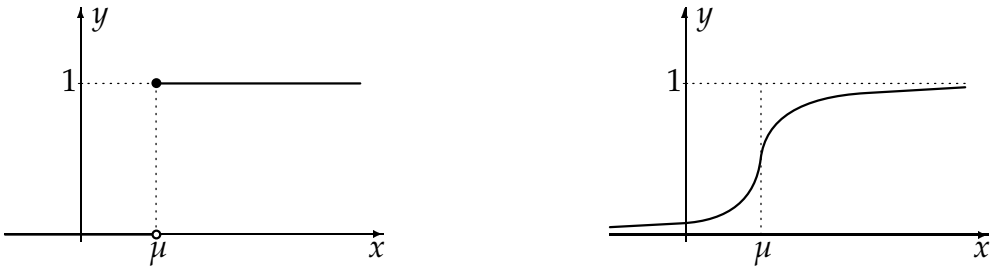

FIGURE 3.3: Fonctions de décision : une fonction à échelon et sigmoïde.

biologique, mais contient déjà tout les concepts importants. Le neurone de McCulloch et Pitts est un système avec plusieurs entrées u_j (cf. dendrites) et une sortie y (cf. axone). Pour déterminer la sortie, une somme pondérée (cf. influence synapses) des entrées est comparée avec un seuil μ. On peut écrire ce modèle mathématiquement comme suit.

$$x = \sum_j a_j u_j = \boldsymbol{a} \cdot \boldsymbol{u},$$
$$\begin{cases} y = +1 & \text{si } x \geq \mu, \\ y = 0 & \text{si } x < \mu. \end{cases} \tag{3.1}$$

Les coefficients a_j sont les poids synaptiques et (3.1) est la *fonction de décision*. Une valeur de $+1$ pour y correspond à la génération d'un potentiel d'action.

On peut généraliser ce modèle en permettant des fonctions de décision plus générales, $y = f(x)$. Par exemple, une fonction couramment utilisée pour modéliser un neurone réel est la fonction sigmoïde. Une représentation schématique du modèle est donnée dans la figure 3.2. Deux exemples de fonctions de décision sont données dans la figure 3.3.

Les poids synaptiques sont déterminés par une phase d'apprentissage. La règle qualitative pour guider l'apprentissage proposée dans [7] par Hebb en 1949 connaît une utilisation très répandue. Cette *règle de Hebb* dit qu'on doit accroître la valeur des poids synaptiques entre neurones formels qui ont une activité synchronisée (i.e. un neurone persiste à aider l'autre à générer un potentiel d'action).

3.2.3 Réseaux de neurones artificiels

L'évolution dans le domaine de la modélisation des réseaux de neurones et la puissance de ceux-ci pour le traitement de l'information ont été responsable de la naissance d'un nouveau domaine de recherche : les *réseaux de neurones artificiels*.

Ces systèmes sont construits avec les propriétés et points forts des réseaux de neurones biologiques en tête. Les applications comprennent bien sûr la simulation des réseaux de neurones biologiques et l'intelligence artificielle, mais aussi une gamme très étendue de différents types de traitement de données. Cette gamme comprend entre autres la classification, la reconnaissance de formes, l'approximation de fonctions et la modélisation de systèmes (linéaires et non linéaires).

Souvent le caractère neuronal de beaucoup de méthodes nées dans le domaine des réseaux de neurones artificiels n'est plus visible. Elles sont devenues des méthodes mathématiques, souvent sous forme d'algorithmes, de traitement de données. Plusieurs de ces méthodes mathématiques peuvent aussi être construites dans d'autres cadres que celui des réseaux de neurones.

En résumant, on pourra dire que la façon dont les organismes résolvent des problèmes de traitement de données dans leur environnement naturel nous a donné de l'inspiration pour construire des méthodes de traitement de données pour les problèmes que nous rencontrons.

3.2.4 Modèle fonctionnel

Toutes les méthodes neuronales sont construites à partir d'un modèle fonctionnel, qui est un des types de réseaux de neurones artificiels. Comme les réseaux de neurones biologiques, ce système comprends des éléments (les *neurones formels*), un schéma d'interconnexion (l'*architecture*) et une procédure de sélection des paramètres du modèle (les *règles d'apprentissage*).

Les neurones formels sont des systèmes dont le modèle est montré dans la figure 3.2. Les entrées u_i sont pondérées (avec les a_i) et intégrées pour donner l'état interne x. La transformation f de cet état interne donne la sortie $y = f(x)$.

3.2.4.1 Neurones formels

Les deux grandes classes de neurones formels sont les neurones linéaires (avec f la fonction d'identité : $f(x) = x$) et les neurones non linéaires (avec f une fonction non linéaire : échelon, sigmoïde,...).

3.2.4.2 Architecture

L'architecture est décrite par un graphe orienté avec les neurones formels comme nœuds. Les connexions correspondent aux entrées externes et les sor-

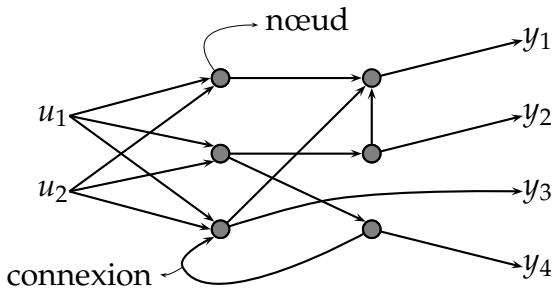

FIGURE 3.4: Illustration de l'architecture.

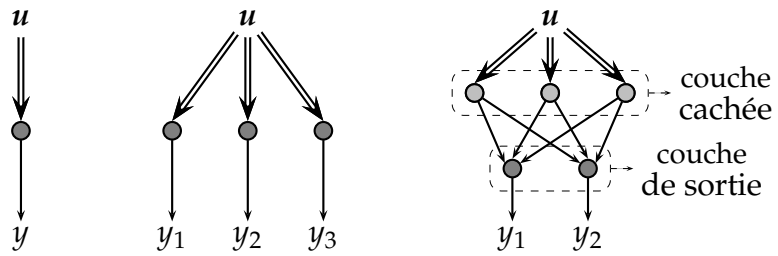

FIGURE 3.5: Trois architectures : le neurone simple, le réseau mono-couche et le réseau multi-couches.

ties des neurones formels (qui peuvent être des entrées pour les neurones formels). Un exemple illustratif est donné dans la figure 3.4.

On peut distinguer trois grands types d'architectures couramment utilisées. Le plus simple est le *neurone simple*, une architecture avec une ou plusieurs entrées et une seule sortie. Les réseaux à une couche (*mono-couche*) ont plusieurs sorties et il n'y a en général pas de connexions entre les neurones de la couche. Les réseaux à couches multiples (*multi-couches*) consistent en au moins deux couches, dont celles qui n'ont pas de sorties vers l'extérieur sont appelées cachées. Ces trois types sont illustrés dans la figure 3.5. Les connexions correspondant aux les entrées $\boldsymbol{u} = (u_1, \ldots, u_n)$ sont remplacées par une double flèche. En général, les neurones d'une couche sont du même type (ont la même fonction de décision f). Comme on peut toujours trouver un réseau mono-couche linéaire (à neurones linéaires) qui donne les même sorties qu'un réseau multi-couches linéaire, nous supposerons que les réseaux multi-couches ont au moins une couche non linéaire.

3.2.4.3 Règles d'apprentissage

Les règles d'apprentissage peuvent agir en général sur l'ensemble des paramètres du modèle fonctionnel. Dans les modèles couramment utilisés, ce sont surtout les poids qui sont adaptés pendant la phase d'apprentissage. Parfois l'architecture est modifiée (réseau à structure évolutive), mais la fonction

de décision f n'est presque jamais modifiée. La majorité des règles d'apprentissage utilisées repose sur la règle de Hebb et opère à partir d'une base de données d'apprentissage.

Pour discuter les règles d'apprentissage des poids,[1] il est utile de savoir en quoi consiste la base de données d'apprentissage. Cette base de données consiste en une liste d'entrées $\boldsymbol{u}$ et de sorties $\boldsymbol{y} = (y_1, \ldots, y_m)$ correspondantes. Cette liste peut être disponible dans son entièreté ou consister de couples $(\boldsymbol{u}, \boldsymbol{y})$ successivement disponibles.

Si on a à faire avec un réseau mono-couche, on peut trouver les poids d'un neurone en minimisant un critère d'erreur qui est fonction des poids associés à ce neurone. C'est à dire

$$\begin{aligned} \boldsymbol{a}^* &= \operatorname*{argmin}_{\boldsymbol{a} \in \mathbb{R}^n} \varepsilon(\boldsymbol{a}), \\ \varepsilon(\boldsymbol{a}) &= g(Y, \hat{Y}_{\boldsymbol{a}}), \end{aligned}$$

avec

- g une fonction réelle quelconque,
- Y l'ensemble des sorties y voulues,
- $\hat{Y}_{\boldsymbol{a}}$ l'ensemble des sorties $\hat{y}(\boldsymbol{a}) = f(\boldsymbol{a} \cdot \boldsymbol{u})$ du réseau correspondant aux poids $\boldsymbol{a}$.

Un choix très populaire de critère d'erreur est l'*erreur quadratique moyenne*

$$\varepsilon_{\mathrm{QM}}(\boldsymbol{a}) \propto \sum_{(\boldsymbol{u}, y)} [y - f(\boldsymbol{a} \cdot \boldsymbol{u})]^2.$$

Si on à a faire avec un réseau multi-couches, il faut construire un critère d'erreur $\varepsilon(\boldsymbol{a}, \ldots)$ qui est fonction des poids de tout les neurones. Maintenant, on ne peut plus déterminer les poids optimaux $\boldsymbol{a}^*$ de chacun des neurones à part. Ceci donne par exemple comme erreur quadratique moyenne

$$\varepsilon_{\mathrm{QM}}(\boldsymbol{a}, \ldots) \propto \sum_{(\boldsymbol{u}, \boldsymbol{y})} \|\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{a}, \ldots)\|^2,$$

où $\|\cdot\|$ est une norme quelconque et $\hat{\boldsymbol{y}}(\boldsymbol{a}, \ldots)$ est le vecteur de sortie du réseau.

Considérons d'abord le cas d'un réseau mono-couche. Si les neurones sont linéaires et si la base de données est entièrement disponible, on peut trouver les poids optimaux $\boldsymbol{a}^*$ avec des *méthodes algébriques*, c'est à dire, en résolvant un système linéaire. Cependant, dans la plupart des cas on utilise les *méthodes de gradient*. Ces méthodes itératives venant du domaine de l'optimisation peuvent s'appliquer dans différentes situations. Quand la base de données est entièrement disponible, la règle d'itération s'écrit

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k - \alpha \nabla \varepsilon(\boldsymbol{a}_k),$$

1. On ne considère ici que les règles d'apprentissage pour les poids.

où k désigne l'étape d'itération et $\alpha \in \mathbb{R}$ le pas de gradient. Pour initialiser l'itération, il faut choisir $\boldsymbol{a}_0$ et le pas d'itération, qu'on pourrait prendre variable selon l'itération (notation α_k). Dans cette *méthode du gradient*, on descend à chaque itération sur la même surface d'erreur. Si on ne dispose que d'un élément $(\boldsymbol{u}, \boldsymbol{y})$ de la base de données à la fois cette surface n'est pas connue et il faut adapter la méthode. On définit un critère d'erreur pour chaque itération,

$$\varepsilon_k(\boldsymbol{a}_k) = g(y, \hat{y}_{\boldsymbol{a}}).$$

Ceci donne

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k - \alpha \nabla \varepsilon_k(\boldsymbol{a}_k)$$

comme règle d'itération.[2] Cette dernière méthode est appelée *méthode du gradient stochastique*.[3] Si on prend par exemple l'erreur quadratique moyenne, on trouve

$$\varepsilon_{\mathrm{QM},k}(\boldsymbol{a}_k) \propto [y - f(\boldsymbol{a}_k \cdot \boldsymbol{u})]^2,$$
$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \alpha[y - f(\boldsymbol{a}_k \cdot \boldsymbol{u})] f'(\boldsymbol{a}_k \cdot \boldsymbol{u}) \boldsymbol{u}.$$

Pour un réseau multi-couches on utilise une généralisation de la méthode du gradient,

$$(\boldsymbol{a}_{k+1}, \ldots) = (\boldsymbol{a}_k, \ldots) - \alpha \nabla \varepsilon_k(\boldsymbol{a}_k, \ldots).$$

Ici, le calcul du gradient de l'erreur $\nabla \varepsilon_k(\boldsymbol{a}_k, \ldots)$ se fait avec la *méthode de rétro-propagation du gradient*. Cette méthode n'est rien autre qu'une application de la règle des dérivées de fonctions composées où ont tient compte de la différence entre les sorties des couches cachées et la couche de sortie.

3.2.5 Exemples de réseaux de neurones artificiels

Il est d'abord intéressant d'examiner quelques réseaux classiques.

L'Adaline[4] est un modèle d'un neurone simple linéaire avec la règle du gradient stochastique avec erreur quadratique moyenne. La règle de l'Adaline est donc

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \alpha[y - \boldsymbol{a}_k \cdot \boldsymbol{u}]\boldsymbol{u}.$$

Le Perceptron[5] est un modèle d'un neurone simple non linéaire ($f = \mathrm{sgn}$) qui utilise une règle d'apprentissage qui ressemble fort la règle de l'Adaline, mais n'est ici pas déduit avec la règle du gradient stochastique (remarquons que le gradient serai 0 ou $+\infty$). La règle du Perceptron est

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \alpha[y - \mathrm{sgn}(\boldsymbol{a}_k \cdot \boldsymbol{u})]\boldsymbol{u},$$

2. Cette règle est appelée *règle delta généralisée*.
3. On a montré que cette méthode, sous certaines conditions de régularité, converge vers la solution de la méthode du gradient, mais plus lentement.
4. ADAptive LINear Element, un modèle proposé par B. Widrow en 1960 dans [8].
5. Un modèle proposé par F. Rosenblatt en 1960 dans [9].

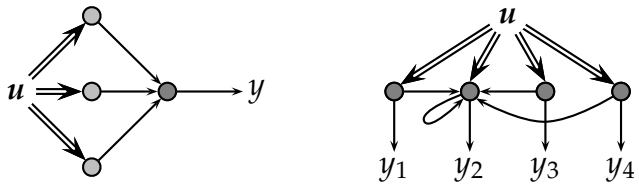

FIGURE 3.6: Schémas des réseaux RBF et SOM.

avec $y \in \{-1, 1\}$ et

$$\text{sgn} : \mathbb{R} \to \{-1, 0, 1\} : x \to \text{sgn}(x) = \begin{cases} -1 & \text{si } x \in \mathbb{R}^-, \\ 0 & \text{si } x = 0, \\ 1 & \text{si } x \in \mathbb{R}^+. \end{cases}$$

La règle du Perceptron est une bonne illustration que le modèle fonctionnel standard de §3.2.4 ne doit pas être considéré comme un dogme.

Les deux neurones précédents et d'autres types peuvent bien sûr être utilisé pour former des réseaux. Par exemple, un réseau populaire est le réseau multi-couches appelé Perceptron multi-couches, qui est constitué de neurones avec une fonction de décision f différentiable (et donc pas celle du modèle du Perceptron même...).

Examinons maintenant les deux types de réseau qu'on utilise dans ce mémoire.

Le premier type est à la base de la construction des cartes auto-organisatrices (SOM[6]) ou cartes de Kohonen, d'après leur inventeur. Une illustration de l'architecture de ce réseau est donnée à droite dans la figure 3.6. C'est un réseau mono-couche, mais avec des connexions additionnelles entre les neurones. Dans la figure, on a uniquement montré les connexions vers un neurone, mais en réalité toutes les sorties sont connectées avec tous les neurones. Une interprétation des connexions sera donnée dans §3.3.4. La règle d'apprentissage consiste en deux étapes. D'abord, un voisinage N est défini autour du neurone, dit gagnant, avec la plus grande sortie. Ce voisinage n'est rien autre qu'un sous-ensemble des neurones du réseau. Les poids des neurones en dehors de N ne changent pas et les poids des neurones dans N sont adaptés comme suit,

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \alpha[\boldsymbol{u} - \boldsymbol{a}_k].$$

On entrera plus dans les détails de la construction des cartes auto-organisatrices dans §3.3, où on les considérera plutôt comme une méthode mathématique.

Le réseau à fonctions radiales de base (réseaux RBF[7]) est le deuxième type de réseau qu'on utilisera dans ce mémoire. L'architecture de ce réseau multicouches est montrée à gauche dans la figure 3.6. Le neurone de sortie, qui du

6. Abréviation venant de l'anglais «Self Organizing Map».
7. Abréviation venant de l'anglais «Radial Basis Functions».

reste ne doit pas être unique, est linéaire. Les neurones de la couche cachée ont une fonction de décision radiale, c'est à dire une fonction qui ne dépend que de la distance $\|\boldsymbol{u} - \boldsymbol{c}_i\|$. Ici, $\boldsymbol{c}_i$ est le vecteur des paramètres d'un neurone, qui correspond aux vecteur $\boldsymbol{a}$ de poids dans les autres réseaux qu'on a déjà vus. La distance euclidienne peut par exemple être choisie pour $\|\cdot\|$. La sortie est donc facile à calculer,

$$y = \sum_i \lambda_i f(\|\boldsymbol{u} - \boldsymbol{c_i}\|),$$

où les λ_i sont les poids du neurone linéaire. L'apprentissage se fait par méthode de gradient avec rétro-propagation du gradient. Ce type de réseau montre bien que dans les réseaux de neurones artificiels on ne doit pas se limiter à une simple pondération des données $\boldsymbol{a}\cdot\boldsymbol{u}$ comme dans les modèles de neurones biologiques avant d'appliquer la fonction de décision. On prendra une vue plus mathématique des réseaux à fonctions radiales de base dans §3.4.

3.3 Cartes auto-organisatrices

Dans cette section on répondra aux questions suivantes. Qu'est-ce qu'une carte auto-organisatrice ? Comment est-ce qu'on construit ces cartes ? Quelles sont ses caractéristiques ? Dans quel but est-ce qu'on pourrait les utiliser ? En plus, on donnera d'autres informations de caractère contextuel ou exemplatif.

3.3.1 Quantification vectorielle

La *quantification vectorielle*[8] classique est une méthode qui fait une approximation d'une fonction de densité de probabilité de vecteurs stochastiques,[9]

$$p : \mathbb{R}^n \to \mathbb{R} : \boldsymbol{u} \to p(\boldsymbol{u}),$$

avec

$$\begin{aligned} \int_{\mathbb{R}^n} p(\boldsymbol{u}) d\boldsymbol{u} = 1 \quad &\text{si } p \text{ est continue,} \\ \sum_i p_i(\boldsymbol{u}) = 1 \quad &\text{si } p \text{ est discrète.} \end{aligned}$$

Cette approximation est faite en utilisant un nombre fini m de *centroïdes*[10] $\boldsymbol{c}_i \in \mathbb{R}^n$. Si ces vecteurs $\boldsymbol{c}_i$ sont choisis, l'approximation est faite en remplaçant

8. En anglais, on dit «vector quantization».
9. Nous prendrons ici $\mathbb{R}^n$ comme l'espace des données, mais on pourra bien choisir d'autres ensembles. Il est clair qu'on pourra prendre un sous-ensemble de $\mathbb{R}^n$, mais par exemple aussi l'espace X^n de mots de n lettres de l'alphabet X. On aura uniquement besoin d'une norme définie sur l'espace des données.
10. En anglais, on dit «codebook vector».

les vecteurs d'entrées par le centroïde le plus proche. C'est à dire,

$$\boldsymbol{c_u} = \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \|\boldsymbol{u} - \boldsymbol{c}_i\|,$$

où la notation $\boldsymbol{c_u}$ est utilisée pour désigner le centroïde qui est le plus proche de $\boldsymbol{u}$.[11] Pour la mesure de distance on peut choisir une norme $\|\cdot\|$ quelconque, mais le choix classique est la norme euclidienne.

La question se pose de la façon de choisir les centroïdes. Dans la quantification vectorielle classique on les définit en minimisant l'erreur ε dite de quantification,

$$\varepsilon = \int_{\mathbb{R}^n} \|\boldsymbol{u} - \boldsymbol{c_u}\|^2 p(\boldsymbol{u}) d\boldsymbol{u},$$

où on a pris la norme euclidienne. Il faut noter que $\boldsymbol{c_u}$ est fonction de $\boldsymbol{u}$ et de tous les $\boldsymbol{c}_i$. Comme ce n'est en général pas possible de trouver une expression explicite pour les centroïdes, on est obligé d'utiliser des méthodes itératives (voir [3, *§1.5.2*]). Pour chaque itération k on tire un $\boldsymbol{u}_k$ au hasard. Si on utilise une méthode de descente de gradient, on trouve le *processus d'apprentissage* suivant

$$\boldsymbol{c}_{i,k+1} = \boldsymbol{c}_{i,k} + \alpha_k \delta_{\boldsymbol{u}i} [\boldsymbol{u}_k - \boldsymbol{c}_{i,k}], \tag{3.2}$$

où $0 < \alpha_k \leq 1$ est *la largeur du pas* choisie et $\delta_{\boldsymbol{u}i}$ le delta de Kronecker qui vaut 1 si $\boldsymbol{c}_{\boldsymbol{u}_k} = \boldsymbol{c}_{i,k}$ et zéro sinon. Les $\boldsymbol{c}_{i,0}$ initiaux peuvent être choisis par tirage au hasard de m vecteurs $\boldsymbol{u}$. On prend α_k entre 0 et 1 pour assurer qu'on prend toujours des combinaisons convexes de $\boldsymbol{c}_{i,k}$ et $\boldsymbol{u}_k$.

Souvent nous n'avons pas de fonction de densité de probabilité p à notre disposition, mais un ensemble fini d'échantillons $\boldsymbol{u}$ d'une densité de probabilité inconnue. Il est clair qu'on peut utiliser ces échantillons pour les $\boldsymbol{u}_k$ dans la règle d'itération.

Si on prend un nombre de tirages $\boldsymbol{u}_k$ assez grand (> 1000) et un pas α_k qui évolue de 1 à un pas beaucoup plus petit que 1 les centroïdes vont converger vers des positions fixes. Si nécessaire, on peut recycler les tirages qui sont à notre disposition. Le nombre de fois qu'on recycle les données disponible est appelé le nombre d'*époques*.

Jusqu'à maintenant on a parlé de la quantification vectorielle classique, mais il est aussi devenu clair qu'on peut la généraliser. Une façon de le faire est de choisir un critère d'erreur ε différent. Une autre façon de le faire est de directement définir un processus d'apprentissage. On verra que les cartes auto-organisatrices ne sont rien d'autre qu'une méthode de quantification vectorielle. Pour ces généralisations, le but n'est pas toujours d'approcher $p(\boldsymbol{u})$, mais de «résumer» les données d'entrées $\boldsymbol{u}$ d'une façon inspirée par le but de la quantification vectorielle.

---

11. Si nécessaire, on pourrait par exemple associer un $\boldsymbol{c_u}$ aux vecteurs $\boldsymbol{u}$ qui sont aussi proches de différentes centroïdes en choisissant le $\boldsymbol{c}_i$ sur base de l'indice i.

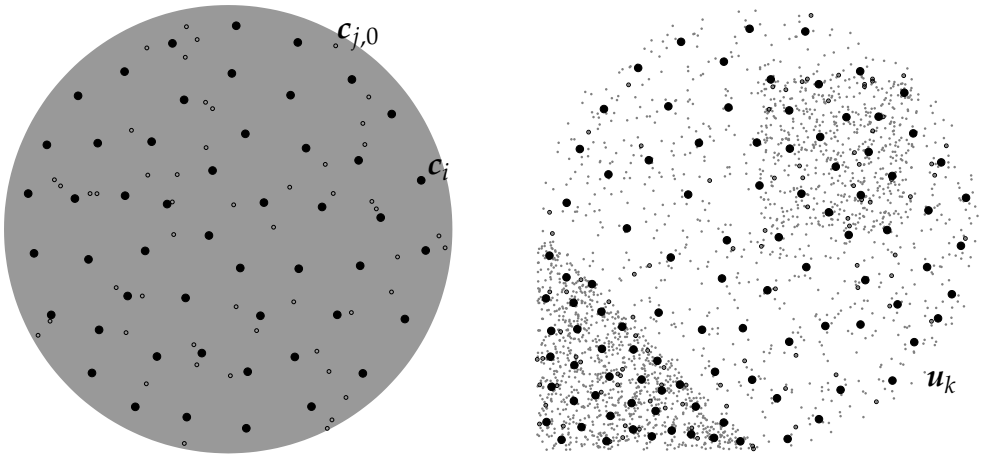

FIGURE 3.7: Illustration de la méthode de la quantification vectorielle.

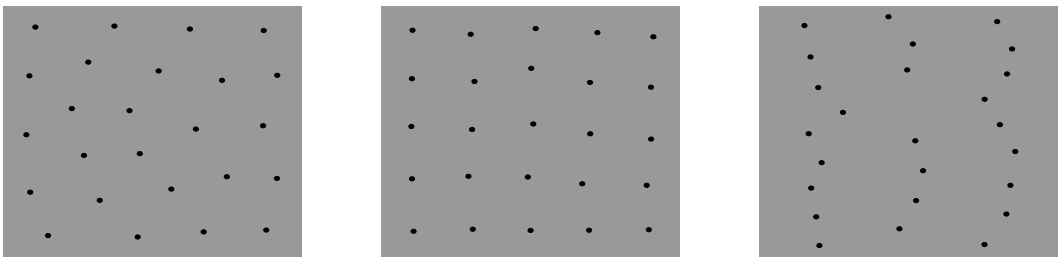

FIGURE 3.8: Quantification vectorielle avec différentes normes.

Dans la figure 3.7 on montre deux exemples dans le plan de la méthode de quantification vectorielle. Les centroïdes finals $\boldsymbol{c}_i$ sont représentés par des boules noires et les petits cercles correspondent aux centroïdes initiaux $\boldsymbol{c}_{i,0}$. A gauche on montre le résultat d'une quantification d'une distribution uniforme (montrée en gris). A droite on montre une quantification d'une distribution non uniforme, pour laquelle les tirages $\boldsymbol{u}_k$ sont représentés par des petits cercles en gris.

Dans la figure 3.8 nous montrons encore trois quantifications d'une même distribution (rappelons que le fond gris représente une distribution uniforme), mais avec trois normes différentes.[12] On l'a fait pour illustrer l'influence de la norme utilisée sur les positions finales des centroïdes.

12. De gauche à droite on à utilisé
- la norme euclidienne (comme avant), $\|\boldsymbol{u}\| = \sqrt{\sum_{i=1}^{n} u_i^2}$ avec un cercle comme lieu de distance constante (dans le plan) ;
- la norme du supremum, $\|\boldsymbol{u}\|_\infty = \sup_i u_i$ avec un carré comme lieu de distance constante ;
- une norme de fantaisie, $\|\boldsymbol{u}\|_* = \sqrt{\sum_{i=1}^{n} i^{2i} u_i}$ avec un ellipse comme lieu de distance constante.

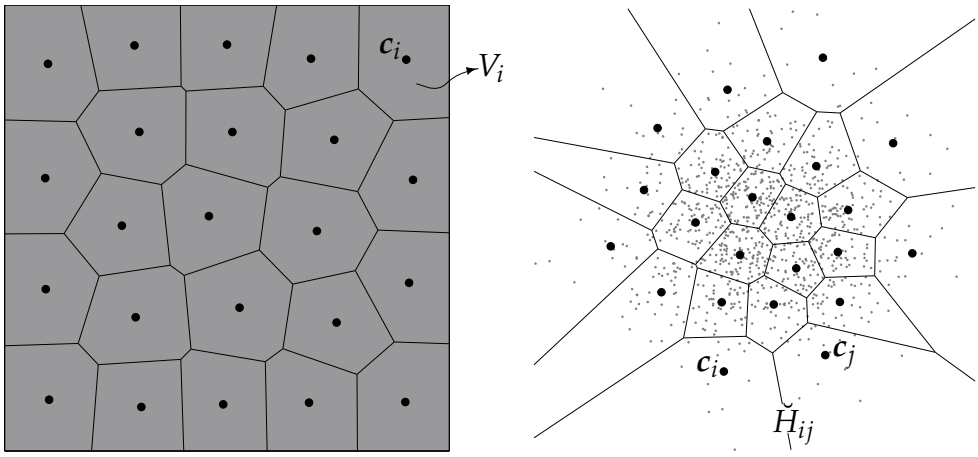

FIGURE 3.9: Illustration de la partition en zones de Voronoi.

Les méthodes de quantification vectorielle peuvent par exemple être utilisées pour faire de la classification. Une autre application est de limiter le temps de calcul, spécifiquement pour transmettre des valeurs de signaux. Au lieu de transmettre la vraie valeur $\boldsymbol{u}$, on utilise l'indice du centroïde $\boldsymbol{c}_i$.

3.3.2 Partition en zones de Voronoi

Un concept très utile dans le cadre de la quantification vectorielle est la *partition en zones de Voronoi*[13]. La partition en zones de Voronoi divise l'espace $\mathbb{R}^n$, ou le sous espace relevant, en régions délimitées par des parties d'hyperplans. Ces hyperplans H_{ij} correspondent au points de $\mathbb{R}^n$ qui sont aussi proches de deux centroïdes $\boldsymbol{c}_i$ et $\boldsymbol{c}_j$, c'est à dire

$$H_{ij} = \{\boldsymbol{u} \mid \|\boldsymbol{u} - \boldsymbol{c}_i\| = \|\boldsymbol{u} - \boldsymbol{c}_j\|\}.$$

On ne garde que les parties suivantes des hyperplans

$$\breve{H}_{ij} = \{\boldsymbol{u} \subset H_{ij} \mid \|\boldsymbol{u} - \boldsymbol{c}_i\| \leq \|\boldsymbol{u} - \boldsymbol{c}_l\|, l \neq i, j\}.$$

L'ensemble de ces $\breve{H}_{ij}$ constitue la partition.

L'espace est donc divisé en régions V_i, dites *zones de Voronoi*, autour des centroïdes $\boldsymbol{c}_i$ telles que les vecteurs $\boldsymbol{u}$ dans V_i sont plus proches de $\boldsymbol{c}_i$ que de tous les autres centroïdes.

Dans la figure 3.9 on donne une illustration de ce qui précède. A gauche on a montré la partition en zones de Voronoi pour une quantification vectorielle d'une distribution uniforme et à droite pour une distribution gaussienne.

13. En anglais, on dit «Voronoi tesselation».

3.3.3 Cartes auto-organisatrices

Les *cartes auto-organisatrices* sont une classe spécifique de méthodes de quantification vectorielle.[14] Le processus d'apprentissage (3.2) est maintenant

$$\boldsymbol{c}_{i,k+1} = \boldsymbol{c}_{i,k} + \alpha_k h_k[\boldsymbol{u}_k - \boldsymbol{c}_{i,k}], \tag{3.3}$$

où la fonction h_k remplace le delta de Kronecker $\delta_{\boldsymbol{u}i}$. Nous rappelons que $\boldsymbol{c}_{\boldsymbol{u}}$ désigne le centroïde le plus proche de $\boldsymbol{u}_k$. Dans la quantification vectorielle classique on ne déplaçait à chaque itération que $\boldsymbol{c}_{\boldsymbol{u}}$. Ici, on permet que plusieurs centroïdes se déplacent simultanément. C'est dans h_k qu'on définit quels centroïdes peuvent se déplacer et quelle est l'atténuation de la largeur du pas. Dans les méthodes des cartes auto-organisatrices, h_k définit un voisinage du $\boldsymbol{c}_{\boldsymbol{u}}$ pour l'itération k et est donc appelée *fonction de voisinage*.

3.3.3.1 Voisinage

Pour arriver au concept de voisinage, il faut d'abord associer une *structure* à l'ensemble des centroïdes.[15] Cette structure associe à chaque centroïde $\boldsymbol{c}_i \in \mathbb{R}^n$ une position $\boldsymbol{r}_i \in \mathbb{R}^s$ où $1 \leq s \leq n$. Normalement ces positions sont choisies telle que l'ensemble des $\boldsymbol{r}_i$ forme une structure régulière comme une ficelle ($s = 1$), une grille rectangulaire ou hexagonale ($s = 2$). Généralement, on considère uniquement les cas à une dimension et deux dimensions. Ceci parce que les cartes auto-organisatrices dans ces dimensions sont très pratique pour faire des représentations visuelles de données à grande dimension $n > 2$.[16]

Pour bien montrer la structure d'un ensemble de centroïdes, on utilisera l'indexation $\boldsymbol{c}_i = \boldsymbol{c}_{\boldsymbol{r}_i}$. Pour la ficelle et pour la grille rectangulaire, on peut limiter les positions à $\mathbb{Z}^s$, ce qui peut en outre beaucoup faciliter les calculs.

Pour la quantification vectorielle classique, les indices i des $\boldsymbol{c}_i$ ne servaient que pour l'identification des centroïdes. Les indices $\boldsymbol{r}_i$ par contre sont utilisés pour définir les voisinages. La fonction de voisinage

$$h_k : \mathbb{R}^s \times \mathbb{R}^s \to [0,1] : (\boldsymbol{r}_i, \boldsymbol{r}_{\boldsymbol{u}}) \to h_k(\boldsymbol{r}_i, \boldsymbol{r}_{\boldsymbol{u}}),$$

peut être choisie dans deux grandes classes.

Il y a d'abord les *voisinages abrupts*. Dans ce cas, $h_k \in \{0,1\}$ et le voisinage est constitué des centroïdes $\boldsymbol{c}_i$ pour lesquels $h_k(\boldsymbol{r}_i, \boldsymbol{r}_{\boldsymbol{u}}) = 1$. Des exemples de ces voisinages abrupts sont les voisinages linéaire, rectangulaire et hexagonal. On les a illustré (pour deux instants k) dans la figure 3.10. Là-dessus sont dessinées

14. Pour une explication du lien entre la quantification vectorielle classique et les cartes auto-organisatrices, voir [3, *§3.12, §3.13*].
15. On pourrait associer le réseau de neurones qu'on a vu dans §3.2.5 avec cette structure et les centroïdes avec les poids des neurones.
16. Parfois, $s = 3$ est aussi utilisée dans des cas spécifiques où les $\boldsymbol{r}_i$ sont sur une surface déroulable (comme un cylindre) ou dépliable (comme un cube).

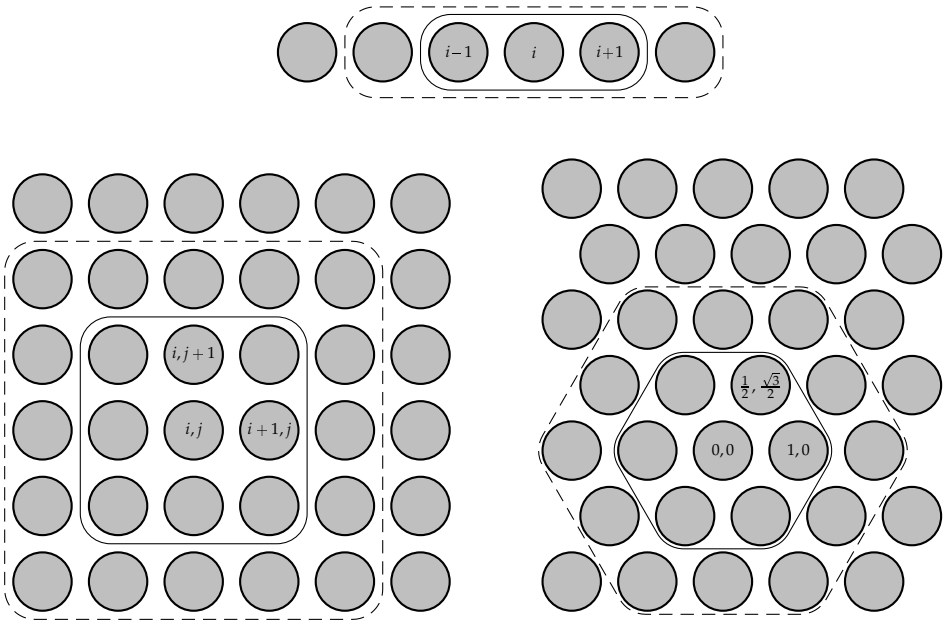

FIGURE 3.10: Trois structures de centroïdes et exemples de voisinages.

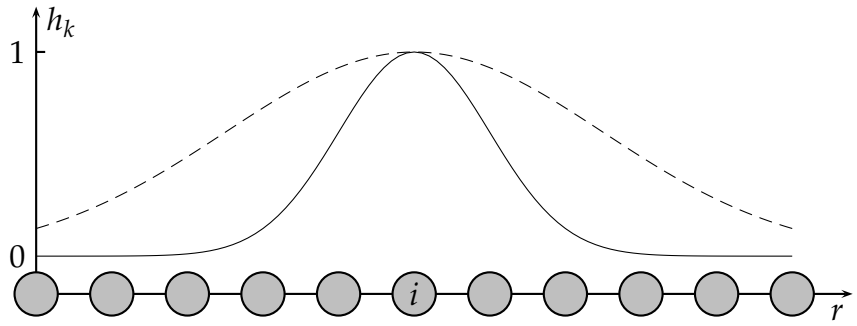

FIGURE 3.11: Exemple d'un voisinage lisse.

des structures de la ficelle (en haut), de la grille rectangulaire (à gauche) et de la grille hexagonale (à droite). Ces structures sont représentées par des ensembles de cercles ordonnés (correspondants aux centroïdes). Pour quelques centroïdes on a aussi donné le $\boldsymbol{r}_i$ correspondant.

L'autre classe est celle des *voisinages lisses*. Dans ce cas, $h_k \in [0, 1]$ et selon la valeur de $h_k(\boldsymbol{r}_i, \boldsymbol{r}_u)$ les centroïdes $\boldsymbol{c}_i$ se trouvent plus ou moins dans le voisinage du centroïde $\boldsymbol{c}_u$. L'exemple classique est un voisinage gaussien,

$$h_k(\boldsymbol{r}_i, \boldsymbol{r}_u) = \exp(-\frac{\|\boldsymbol{r}_u - \boldsymbol{r}_i\|^2}{2\sigma_k^2}),$$

où σ_k est la «largeur» du voisinage. Dans la figure 3.11 on a montré un tel voisinage gaussien pour une ficelle, avec $\sigma_k = 1$ et $\frac{5}{2}$.

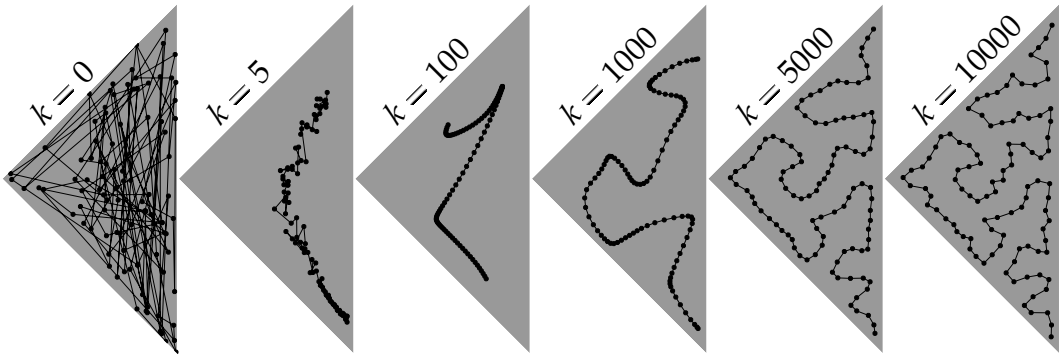

FIGURE 3.12: Évolution des centroïdes avec k.

3.3.3.2 Convergence

On voit qu'il y a une diversité énorme dans les cartes auto-organisatrices. On peut choisir la structure, le type de voisinage et les normes utilisées dans l'espace des données $\mathbb{R}^n$ et l'espace de la structure $\mathbb{R}^s$ (ces normes sont généralement des normes euclidiennes). Ces choix doivent être faits avec l'application de la carte auto-organisatrice en tête. Si on a décidé des choix ci-dessus, il faut choisir la largeur de pas α_k et la fonction de voisinage h_k pour tout les k. Il n'y a pas de règles fixes pour le faire, mais il faut prendre un α_k et une largeur de voisinage (définie par h_k) monotone décroissante. La largeur de voisinage (la cardinalité d'un ensemble de centroïdes pour les voisinages abrupts et un paramètre comme σ_k pour les voisinages lisses) doit commencer (k petit) très grand, contenant plus que la moitié des centroïdes, et diminuer jusqu'à ce qu'on ne garde que le centroïde $\boldsymbol{c_u}$, éventuellement avec ses plus proches voisins.

Le résultat de la méthode de quantification vectorielle décrite ci-dessus est le suivant. Les centroïdes qui sont topographiquement proches dans la structure (dans $\mathbb{R}^s$) seront topographiquement proches dans l'espace des données ($\mathbb{R}^n$). Ceci veut dire que la structure sera reflétée dans l'ordre que prennent les centroïdes dans l'espace des données. On discutera les caractéristiques des quantifications obtenues à l'aide de quelques exemples. Pour montrer la structure, on a dessiné des lignes entre plus proches voisins dans les illustrations qui viennent.

D'abord, regardons l'évolution des centroïdes avec l'itération k. Dans la figure 3.12 on a pris une ficelle avec 100 centroïdes comme structure et un voisinage gaussien. Les données $\boldsymbol{u}$ viennent d'une distribution uniforme et triangulaire. Les valeurs initiales étaient choisies aléatoirement. On voit qu'initialement ($k \leq 1000$) la structure se forme et puis les itérations servent principalement pour approcher la distribution. Notons que l'ordre final ressemble à une courbe fractale.

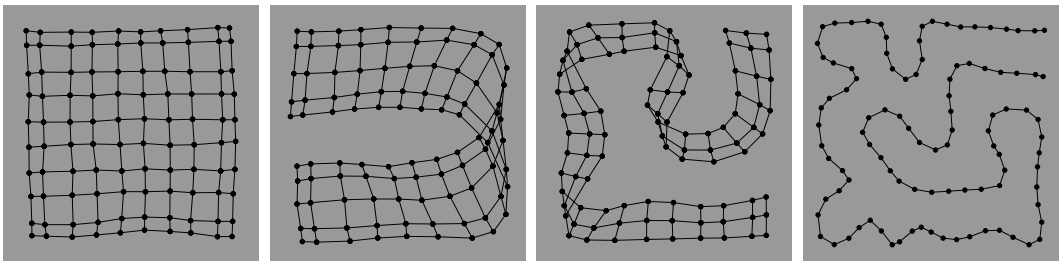

FIGURE 3.13: Cartes illustratrices de quelques caractéristiques.

Dans l'exemple précédent, on voit qu'il y a convergence vers un ensemble de centroïdes ordonnés. Il est clair que le choix des α_k et h_k détermine s'il y a «bonne» convergence ou pas. Pour l'exemple de la figure 3.12 il est clair qu'on veut éviter que les centroïdes restent plus ou moins fixés dans leurs positions de $k = 0, 5, 10, 100, 1000$. Malheureusement, il y a peu de résultats mathématiques sur la convergence de cartes auto-organisatrices qui peuvent nous aider. On doit alors utiliser des règles venant de l'expérimentation (voir par exemple [3, *§3.1, §3.6, §3.7*]).

3.3.3.3 Caractéristiques

A partir des cartes[17] dans la figure 3.13 on peut discuter quelques caractéristiques des cartes après convergence. Ces cartes comprennent à peu près 100 centroïdes et sont construites avec des voisinages abrupts ; la norme est euclidienne. Sur les deux cartes à gauche on voit qu'il y a un effet de bord : les centroïdes se trouvent plus proches l'un de l'autre aux bords de la structure. Cet effet diminue en augmentant le nombre de centroïdes. Les deux cartes au milieu sont pliées sur elles-mêmes. Cet effet se produit quand les dimensions de la structure ne sont pas adaptées à la distribution p. Quand la dimension n de l'espace des données est plus grand que la dimension s de l'espace de la structure, on a beaucoup moins de risque de repliement (ce qu'on peut voir sur la carte à droite). Sur toutes ces cartes les centroïdes voisins dans la structure se trouvent proches dans l'espace des données. Mais ceci ne veut pas nécessairement dire que les centroïdes qui se trouvent loin l'un de l'autre dans la structure se trouvent loin l'un de l'autre dans l'espace des données. Ceci est bien illustré sur la carte à droite. Ce dernier effet ne doit pas être vu comme un défaut, mais on en doit tenir compte quand on utilise une carte auto-organisatrice.

Les cartes auto-organisatrices peuvent aussi être faites pour des distributions plus compliquées que celles déjà montrées. Dans la figure 3.14 on montre

17. On utilisera le même terme *carte auto-organisatrice* pour la représentation des centroïdes dans l'espace des données et la réprensentation des centroïdes sur la structure. Le contexte permettra des distinguer les significations.

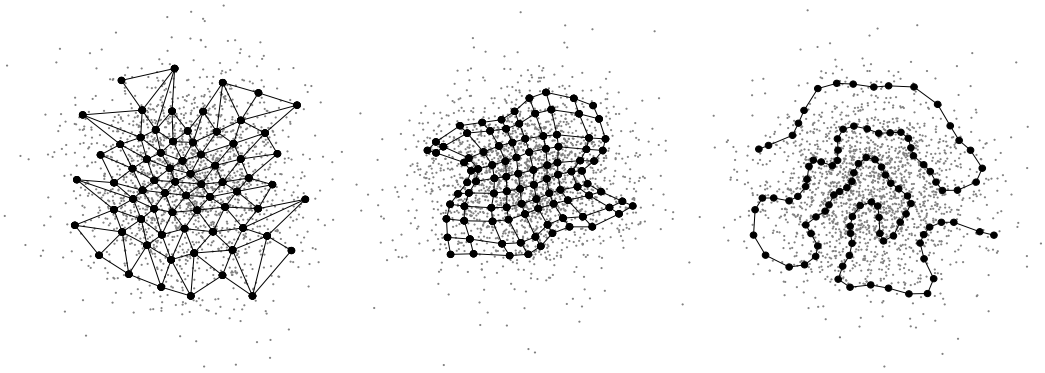

FIGURE 3.14: Cartes faites avec une distribution non-uniforme.

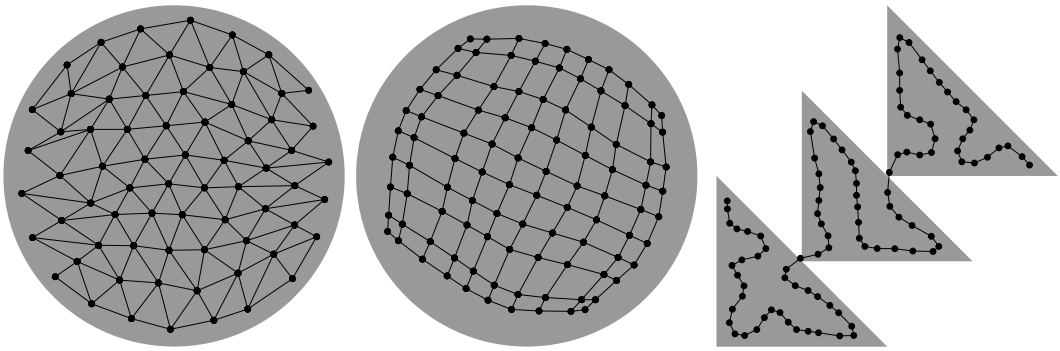

FIGURE 3.15: Encore quelques cartes auto-organisatrices.

trois cartes faites avec une distribution gaussienne à deux dimensions. Dans la figure 3.15 on montre deux cartes faites avec une distribution circulaire uniforme et une carte faite avec une distribution uniforme par parties. Ce qu'on voit dans ces deux figures, c'est qu'avec des structures différentes (et voisinages) on «résume» les distributions p de façons clairement différentes. Laquelle est la meilleure dépend de la distribution et de l'application.

3.3.4 Interprétation comme réseau de neurones

Retournons brièvement au réseau de neurones artificiels à la base des cartes auto-organisatrices. Le schéma de ce réseau a été donné à droite dans la figure 3.6. Nous donnerons une interprétation des différentes éléments du schéma.

Les neurones correspondent aux centroïdes dans la structure ($\boldsymbol{r}_i \in \mathbb{R}^s$).

Les poids des neurones (connexions entrées-neurones) correspondent aux centroïdes dans l'espace des données ($\boldsymbol{c}_i \in \mathbb{R}^n$). On a vu que ces $\boldsymbol{c}_i$ varient pendant la phase d'apprentissage. Contrairement aux connexions précédentes, dites plastiques, les connexions transversales entre neurones sont fixées. Ceci

veut dire que les poids correspondants sont prédéterminés. La fonction de ces connexions est de définir des voisinages de neurones qui vont répondre préférentiellement à un stimulus présenté en entrée. C'est alors ces connexions qui correspondent à la fonction de voisinage h_k, qui est aussi fixée à l'avance.

Il est intéressant de noter qu'on retrouve des structures dans les cerveaux des animaux ressemblant fort à ces cartes auto-organisatrices. Par exemple, des ensembles de neurones pour le traitement des signaux visuels dans lesquels on a des voisinages qui répondent préférentiellement à des signaux tombant sur une partie spécifique de la rétine.

3.3.5 Cartes auto-organisatrices à produit scalaire

Dans cette section on examinera une version légèrement modifiée des cartes auto-organisatrices : les *cartes auto-organisatrices à produit scalaire*[18]. Rappelons qu'on doit déterminer le centroïde $\boldsymbol{c_u}$ qui est le plus proche de l'échantillon $\boldsymbol{u}_k$. On utilisait une norme quelconque pour déterminer $\boldsymbol{c_u}$. Mesurer la distance entre deux points $\boldsymbol{u}$ et $\boldsymbol{v}$ de $\mathbb{R}^n$ n'est rien d'autre que tester leur similarité.

Une autre façon pour tester la similarité de deux points est de considérer le produit scalaire $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$. Nous allons nous limiter ici au produit scalaire classique $\boldsymbol{u}\cdot\boldsymbol{v} = \sum_i^n u_i v_i$. Remarquons que ce produit scalaire engendre la norme euclidienne, $\|u\| = \sqrt{\sum_i^n u_i^2} = \sqrt{\boldsymbol{u}\cdot\boldsymbol{u}}$.

Le centroïde le plus similaire à $\boldsymbol{u}$ est déterminé comme suit,

$$\boldsymbol{c_u} = \underset{\boldsymbol{c}_i}{\operatorname{argmax}}\, \boldsymbol{u}\cdot\boldsymbol{c}_i. \tag{3.4}$$

Il est clair que toutes les données $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ doivent être *normalisées*, c'est à dire $\|\boldsymbol{u}\| = \|\boldsymbol{v}\|$, pour que le produit scalaire puisse être utilisé pour tester la similarité de deux points. Sinon, $\boldsymbol{v} = \alpha\boldsymbol{u}$ avec $\alpha > 1$ serait plus similaire à $\boldsymbol{u}$ que $\boldsymbol{u}$ même. Pour des vecteurs normalisés le critère de similarité (3.4) ne maximise rien autre que $\cos(\widehat{\boldsymbol{u}\boldsymbol{c}_i})$, ou de manière équivalente, minimise l'angle $\widehat{\boldsymbol{u}\boldsymbol{c}_i}$. S'il y a des données avant normalisation avec norme 0, on ne peut pas les normaliser. Mais on peut tenir compte de ces données en ajoutant un centroïde $\boldsymbol{c_0}$ fixé à l'origine. Ce centroïde n'a bien sûr pas de position dans la structure. Il est clair que ces $\boldsymbol{u}_k = \boldsymbol{0}$ seront négligés pendant l'apprentissage.

Dans le cadre décrit ci-dessus le processus d'apprentissage (3.3) devient

$$\boldsymbol{c}_{i,k+1} = \frac{\boldsymbol{c}_{i,k} + \beta_k h_k \boldsymbol{u}_k}{\|\boldsymbol{c}_{i,k} + \beta_k h_k \boldsymbol{u}_k\|},$$

avec $0 < \beta_k < +\infty$ la largeur de pas pour les cartes auto-organisatrices à produit scalaire. Dans ce cas un β_k plus grand que 0 garantit des combinaisons

18. En anglais, on dit «dot-product SOM».

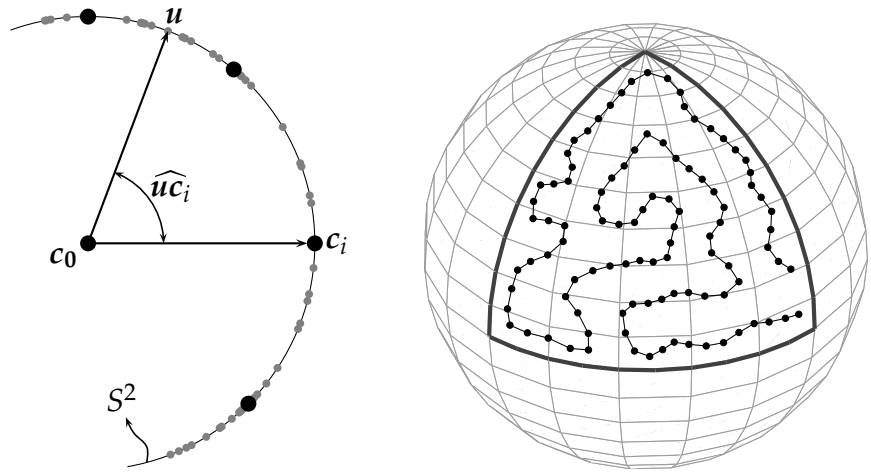

FIGURE 3.16: Illustration des cartes auto-organisatrices à produit scalaire.

convexes de $\boldsymbol{c}_{i,k}$ et $\boldsymbol{u}_k$ qui restent sur l'hypersphère S^n,

$$S^n = \{\boldsymbol{u} \in \mathbb{R}^n \mid \|\boldsymbol{u}\| \text{ est constante}\}.$$

Tout ce qui précède est illustré pour $n = 2$ à gauche dans la figure 3.16. On y trouve aussi un exemple à droite, la distribution est uniforme sur S^3 dans la région délimitée par la ligne épaisse.

3.3.6 Applications des cartes auto-organisatrices

Dans l'introduction de son livre [3], T. Kohonen nous avertit que la méthode des cartes auto-organisatrices n'est pas faite pour la reconnaissance statistique de formes. Ce sont des méthodes de *regroupement*[19], d'abstraction et de *visualisation*.[20] Un deuxième conseil est de ne pas négliger le prétraitement des données.

Une bonne méthode de *prétraitement*, qui n'est pas nécessairement suffisante, est de faire une «normalisation $(0, 1)$» des échelles des propriétés.[21] Nous éclaircissons un peu. Chaque composante u_i de l'entrée $\boldsymbol{u}$ quantifie une certaine propriété. Par exemple, le couple (u_T, u_H) contient des valeurs de température ($u_T \in [20°\text{C}, 100°\text{C}]$) et d'humidité ($u_H \in [0\%, 100\%]$) d'un sauna. Ces deux échelles ont une portée fort différente, l'une entre 20 et 100 et l'autre entre 0 et 1. Pour garantir que les échelles ont la même portée, on peut les normaliser en enlevant la moyenne d'échelle $\mu_i \propto \sum_{\boldsymbol{u}} u_i$ et en divisant par l'écart type de

19. On utilise *regroupement* comme traduction de l'anglais «clustering».
20. Ces méthodes citées sont basées sur le fait que les cartes auto-organisatrices font une *classification non supervisée* de l'espace $\mathbb{R}^n$ (cf. les zones de Voronoi §3.3.2). La reconnaissance statistique de formes est faite avec des méthodes de *classification supervisée*, où les classes sont fixés a priori.
21. On suppose ici que toutes les composantes de $\boldsymbol{u}$ ont la même importance.

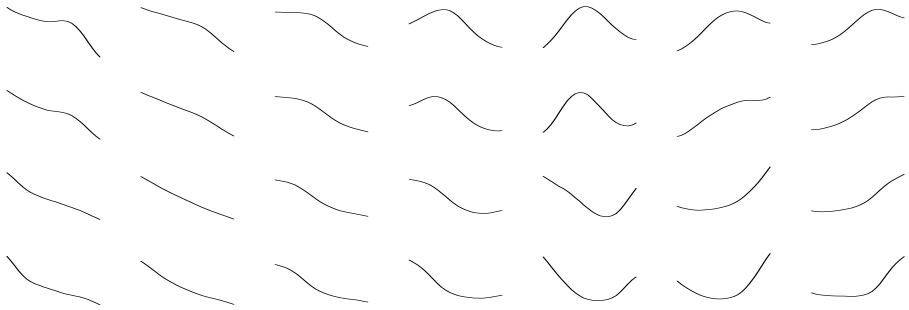

FIGURE 3.17: Exemple de la visualisation des centroïdes.

l'échelle $\sigma_i \propto \sqrt{\sum_{\boldsymbol{u}}[u_i - \mu_i]^2}$ de telle sorte que les échelles normalisées ont une moyenne de 0 et un écart type de 1. Il est clair qu'on pourra aussi bien faire une normalisation qui rend la moyenne et l'écart type identique pour toutes les propriétés. Si on utilise les cartes auto-organisatrices à produit scalaire la «normalisation $(0, 1)$» des échelles des propriétés est faite avant la normalisation des données $\boldsymbol{u}$.

Après le prétraitement on fait le choix du nombre de centroïdes, de la structure et du voisinage. Après apprentissage on utilise la structure pour visualiser les centroïdes. Ainsi, on obtient des cartes auto-organisatrices. Une façon, à côté de beaucoup d'autres, de faire la visualisation est d'afficher les centroïdes comme des courbes $f(i) = u_i$ sur leurs positions $\boldsymbol{r}_i$ dans la structure. Cette méthode est surtout valable si les composantes de $\boldsymbol{u}$ sont des mesures successives d'un certain grandeur. Un exemple d'une telle visualisation pour une structure rectangulaire et avec dimension de données $n = 24$ est montré dans la figure 3.17. Remarquons que les courbes qui représentent les centroïdes sont d'autant plus similaires qu'ils sont plus proches.

Quelques domaines où les cartes auto-organisatrices sont utilisées, sont la reconnaissance de parole, l'analyse de textures, le contrôle d'un robot et l'estimation.

3.4 Réseaux à fonctions radiales de base

3.4.1 Principe

On considérera les *réseaux à fonctions radiales de base* ici comme approximateur de fonctions. Considérons une fonction inconnue

$$g : \mathbb{R}^n \to \mathbb{R} : \boldsymbol{u} \to y = g(\boldsymbol{u}).$$

On approxime cette fonction g en utilisant une famille de *fonctions radiales* ψ_i,

$$\psi_i : \mathbb{R}^n \to \mathbb{R} : \boldsymbol{u} \to \psi_i(\boldsymbol{u}) = \varphi_i(\|\boldsymbol{u} - \boldsymbol{c}_i\|),$$
$$\varphi_i : \mathbb{R}_0^+ \to \mathbb{R} : x \to \varphi_i(x),$$

où $\boldsymbol{c}_i$ est un élément de $\mathbb{R}^n$ appelé centroïde et $\|\cdot\|$ une norme quelconque. L'approximateur $\hat{g} \approx g$ est construit comme une combinaison linéaire de ces fonctions,

$$\hat{g}(\boldsymbol{u}) = \sum_i \lambda_i \varphi_i(\|\boldsymbol{u} - \boldsymbol{c}_i\|). \tag{3.5}$$

Comparons la description des réseaux à RBF d'ici avec celle de §3.2.5. Les centroïdes peuvent être associés avec les neurones de la couche cachée du réseau et la combinaison linéaire avec le neurone de la couche de sortie (cf. figure 3.6).

Un choix typique est de prendre une norme euclidienne et de choisir les gaussiennes multidimensionnelles centrées (sans normalisation) comme famille de fonctions ψ_i. Ceci correspond avec les fonctions réelles suivantes,

$$\varphi_i(\cdot) = \exp(-\frac{1}{2}\left[\frac{\cdot}{\sigma_i}\right]^2).$$

L'écart type σ_i est ici le (seul) paramètre de la famille de fonctions radiales.

Ce qui précède peut facilement être généralisé pour des fonctions $\boldsymbol{g}$ multidimensionnelles,

$$\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{n'} : \boldsymbol{u} \to \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{u}).$$

On ne doit que changer (3.5) pour tenir compte des différentes sorties y_j, c'est à dire

$$\hat{g}_j(\boldsymbol{u}) = \sum_i \lambda_{ij} \varphi_i(\|\boldsymbol{u} - \boldsymbol{c}_i\|).$$

3.4.2 Choix des paramètres

Les premiers choix qui doivent être faits sont la famille de fonctions radiales ψ_i et le nombre de centroïdes $\boldsymbol{c}_i$. Dans la suite on prendra des gaussiennes multidimensionelles comme exemple type de ψ_i et on notera m le nombre de centroïdes.

Pour déterminer les autres paramètres, $\boldsymbol{c}_i$, σ_i et λ_i on utilise un ensemble d'apprentissage T (de taille fini),[22]

$$T = \{(\boldsymbol{u}, y) \in \mathbb{R}^n \times \mathbb{R} \mid y = g(\boldsymbol{u})\}.$$

et un critère d'erreur ε qui dépend de $\boldsymbol{c}_i$, σ_i et λ_i pour tout $i = \{1, \ldots, m\}$.

Même si ces paramètres peuvent être déterminés en utilisant une méthode de descente de gradient comme celles qu'on a vues dans §3.2.4.3, l'algorithme, dit d'entraînement, est souvent découplé en trois étapes. Ces étapes sont

1. déterminer les positions des centroïdes $\boldsymbol{c}_i$,

22. Pour un $\boldsymbol{g}$ multidimensionnelle, on aurait $T = \{(\boldsymbol{u}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^{n'} \mid \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{u})\}$.

2. calculer les largeurs σ_i des fonctions de base,
3. calculer les poids λ_i.

Dans les deux premières étapes seulement les éléments $\boldsymbol{u}$ des couples dans T sont utilisés. Pour la troisième étape les poids λ_i sont calculés en gardant les autres paramètres constants.

Les centroïdes $\boldsymbol{c}_i$ sont calculés en utilisant une méthode de quantification vectorielle (cf. §3.3.1), où l'ensemble d'entrées $\boldsymbol{u}$ est résumé par m centroïdes. Des méthodes très diverses peuvent être utilisées, par exemple la quantification vectorielle classique ou une méthode de carte auto-organisatrice.

Ils existent différentes méthodes pour choisir les largeurs σ_i. Le plus simple est de choisir tout les σ_i égaux à une valeur fixée σ. Un choix qu'on retrouve dans la littérature est de prendre

$$\sigma = \frac{1}{\sqrt{2m}} \max_{i,j} \|\boldsymbol{c}_i - \boldsymbol{c}_j\|.$$

Une deuxième méthode est de choisir une valeur différente σ_i pour chaque fonction radiale de base ψ_i. Cette méthode permet de mieux tenir compte des nonuniformités de la distribution des $\boldsymbol{u}$. On peut par exemple calculer l'écart type σ_{V_i} pour chacun des zones de Voronoi V_i,

$$\sigma_{V_i}^2 = \frac{1}{\sum_{\boldsymbol{u} \in V_i} 1} \sum_{\boldsymbol{u} \in V_i} \|\boldsymbol{u} - \boldsymbol{c}_i\|^2$$

et les multiplier par un facteur constant κ identique pour tout les ψ_i. On obtient donc $\sigma_i = \kappa \sigma_{V_i}$. Le facteur κ peut être déterminé heuristiquement comme suit. On calcule le critère d'erreur ε pour un ensemble de candidats κ_i et on prend celui-ci qui minimise ε,

$$\kappa = \operatorname*{argmin}_{\kappa_i} \varepsilon.$$

Quand les centroïdes $\boldsymbol{c}_i$ et les paramètres sont déterminés, on peut calculer les poids λ_i. Ceci est aussi fait en minimisant ε,

$$\boldsymbol{\lambda} = \operatorname*{argmin}_{\boldsymbol{\lambda} \in \mathbb{R}^m} \varepsilon.$$

Cette façon de déterminer les paramètres est un exemple des méthodes algébriques dont on a parlé dans §3.2.4.3.

Un critère d'erreur couramment utilisé est l'erreur quadratique moyenne,

$$\varepsilon_{\mathrm{QM}} \propto \sum_{(\boldsymbol{u},y) \in T} [y - \hat{g}(\boldsymbol{u})]^2.$$

On voit que le critère est uniquement utilisé pour déterminer le paramètre κ et les poids λ_i. Les autres paramètres $(\boldsymbol{c}_i, \sigma_i)$ ne dépendent que de la distribution des entrées $\boldsymbol{u}$.

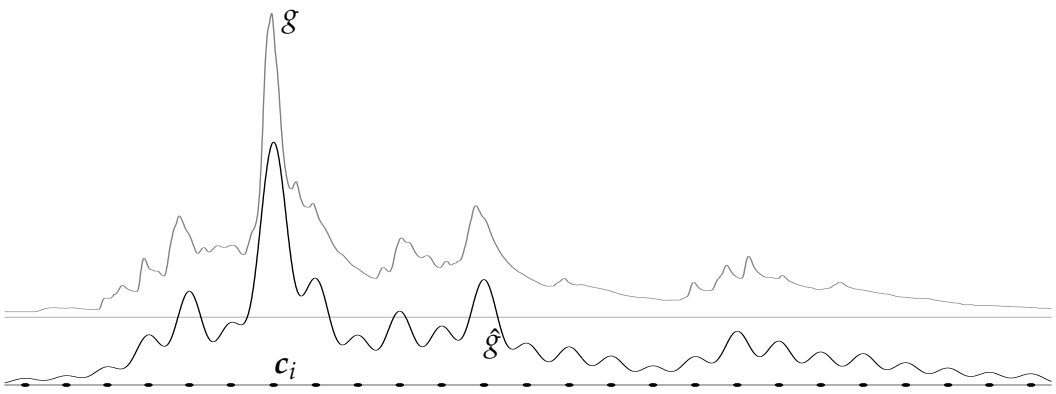

FIGURE 3.18: Exemple d'approximation d'une fonction par un réseaux à fonctions radiales de base.

Dans la figure 3.18 nous avons donné un exemple illustratif d'une approximation $\hat{g}$ d'une fonction g avec 25 centroïdes $\boldsymbol{c}_i$. Pour la clarté, g et $\hat{g}$ sont décalées.

3.5 Bibliographie

La plus grande partie de §3.2 est basée sur [10], qui donne un aperçu du domaine des réseaux de neurones artificiels, comprenant aussi un historique et des applications. Notamment

§3.2.1 sur [10, *Ch.I : I–V*],
§3.2.2 sur [10, *Ch.II : I, II*],
§3.2.4 sur [10, *Ch.II : III ; Ch.III : III–V, VII–IX*] et
§3.2.5 sur [10, *Ch.III : V, VI, IX, X ; Ch.IV : I*].

Pour §3.2.5 on a aussi utilisé [3, *§3.1, §4.1*].

Pour §3.3 la référence de choix était [3], un livre de l'inventeur des cartes auto-organisatrices qui comprend un traitement d'un éventail d'aspects liés aux cartes auto-organisatrices. Plus spécifiquement, on s'est basé pour

§3.3.1 et §3.3.2 sur [3, *§1.5.1, §5.1*],
§3.3.3 sur [3, *§3.1, §3.4, §3.7*],
§3.3.5 sur [3, *§3.2*] et
§3.3.6 sur [3, *§3.10, §7*].

Pour la façon de visualiser montrée dans la figure 3.17 on s'est inspiré sur [11], [12], [13], [14] et [2].

La section §3.4 à été écrite à l'aide de [15]. En plus, quelques parties de [16] ont été utiles pour avoir une meilleure compréhension des RBF. Les références de base pour les RBF sont [17] et [18].

Chapitre 4

Modèles de régression

4.1 Introduction

Dans ce chapitre on examinera des types de modèles qui peuvent être utilisés pour faire des estimations. D'abord, nous examinons les caractéristiques générales de tels modèles, avec une attention spécifique pour la façon de déterminer les paramètres du modèle et l'erreur commise par le modèle. Ensuite, deux modèles spécifiques seront examinés dans plus de détail, un linéaire (ARX) et un non linéaire (basé sur un réseau à RBF).

4.2 Modèles paramétriques

4.2.1 Types de modèles

Il y a deux grand types de modèles, les modèles faits spécifiquement pour un certain système et les modèles boîte-noire.

Les modèles spécifiques sont construits en utilisant des principes physiques dirigeant le système et les paramètres représentent des valeurs inconnues de grandeurs qui ont une interprétation physique. Un exemple d'un tel modèle est la fonction de production dans le modèle Hydromax (voir §5.2.1).

Les modèles du type *boîte-noire* sont des familles de modèles flexibles et très généralement applicables. Ici, les paramètres n'ont pas une interprétation physique, mais caractérisent les propriétés entrée-sortie du système. Des exemples sont les modèles de régression utilisés pour la prédiction de débit dans Hydromax (voir §5.2.2) et pour la prédiction des moyennes, et des écart types dans notre modèle (voir §5.3.2). Nous distinguerons les modèles de régression linéaires et non linéaires.

On note $y_k \in \mathbb{R}$ la sortie du système pour l'instant $k \in \mathbb{Z}$. L'estimation pour cet instant, faite par un modèle paramétrique avec vecteur de paramètres $\boldsymbol{\vartheta}$, est notée $\hat{y}_k(\boldsymbol{\vartheta})$. Le but est bien sûr que les $\hat{y}_k(\boldsymbol{\vartheta})$ soient des bonnes «estimations» de y_k. Pour atteindre ce but on a la liberté de choisir un modèle et les

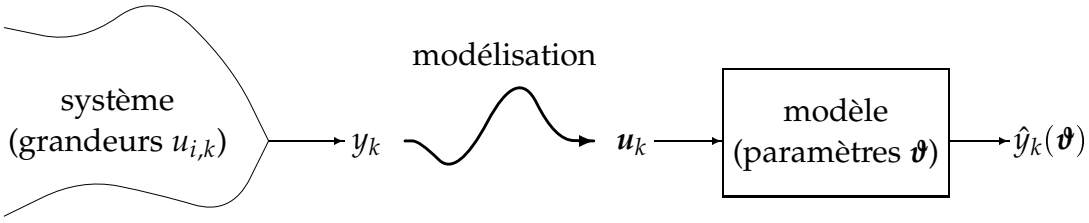

FIGURE 4.1: Modélisation d'un système.

paramètres correspondants $\boldsymbol{\vartheta}$.

Pour le modèle, on a besoin d'entrées $\boldsymbol{u}_k \in \mathbb{R}^l$ disponible à l'instant k. Ces l valeurs correspondent en général avec des mesures de grandeurs caractérisant le système, mais il se peut qu'un $u_{i,k}$ ne correspond pas avec une mesure.

Dans la figure 4.1, on montre les concepts introduit ci-dessus.

Pour les modèles spécifiques, les paramètres peuvent parfois être déterminés par des méthodes conventionnelles de mesure et d'expérimentation. Cependant en général, on utilisera des mesures des entrées et des sorties du système pour estimer les paramètres. Ceci est fait en minimisant un *critère d'erreur* qui quantifie la qualité des prédictions sortant du modèle, si on les compare avec les mesures.

4.2.2 L'erreur associée à un modèle

La valeur du critère d'erreur manié dépendra alors du choix des paramètres du modèle et on la note $\varepsilon(\boldsymbol{\vartheta}) \in \mathbb{R}$. Le critère est défini comme suit

$$\varepsilon(\boldsymbol{\vartheta}) = f(Y, \hat{Y}(\boldsymbol{\vartheta})),$$

avec

- f une fonction réelle quelconque définissant le critère d'erreur,
- $Y = \{y_k \mid k \in I\}$ des mesures venant du système,
- $\hat{Y}(\boldsymbol{\vartheta}) = \{\hat{y}_k(\boldsymbol{\vartheta}) \mid k \in I\}$ des estimations venant du modèle avec vecteur de paramètres $\boldsymbol{\vartheta}$, obtenu à partir des entrées $\boldsymbol{u}_k \in U$,
- $I \subset \mathbb{Z}$ une intervalle de temps.

Un critère d'erreur qui est très souvent utilisé est l'erreur quadratique moyenne

$$\varepsilon_{\mathrm{QM}}(\boldsymbol{\vartheta}) = \frac{1}{n} \sum_{k \in I} [y_k - \hat{y}_k(\boldsymbol{\vartheta})]^2, \tag{4.1}$$

avec n le nombre d'éléments dans I.

Si on utilise l'erreur quadratique moyenne, on peut faciliter les comparaisons de modèles pour différents ensembles de données en faisant une normalisation.[1] La normalisation qu'on utilisera est obtenue en divisant par la

1. Remarquez qu'un changement d'unités, prenons par exemple de mètres à centimètres, a pour effet que l'erreur comme définie dans (4.1) est augmentée d'un facteur 10^4.

variance des données de validation. Ce qui donne comme critère d'erreur

$$\varepsilon_{\overline{\mathrm{QM}}}(\boldsymbol{\vartheta}) = \frac{1}{n\sigma_y^2} \sum_{k \in I} [y_k - \hat{y}_k(\boldsymbol{\vartheta})]^2, \tag{4.2}$$

avec

$$\sigma_y^2 = \frac{1}{n} \sum_{k \in I} [y_k - \mu_y]^2,$$
$$\mu_y = \frac{1}{n} \sum_{k \in I} y_k.$$

4.2.3 Apprentissage : estimation des paramètres

Pour déterminer les paramètres du modèle on utilise des mesures de la sortie du système. Cet ensemble $Y_A = \{y_k \mid k \in I_A\}$ est appelé l'ensemble d'apprentissage. L'ensemble des sorties du modèle correspondant est $\hat{Y}_A(\boldsymbol{\vartheta}) = \{\hat{y}_k(\boldsymbol{\vartheta}) \mid k \in I_A\}$.

Alors, le vecteur des paramètres optimal est

$$\boldsymbol{\vartheta}^* = \operatorname*{argmin}_{\boldsymbol{\vartheta}} \varepsilon(\boldsymbol{\vartheta}),$$

avec $\varepsilon(\boldsymbol{\vartheta}) = f(Y_A, \hat{Y}_A(\boldsymbol{\vartheta}))$ par exemple égal à $\varepsilon_{\overline{\mathrm{QM}}}$. Nous appellerons l'erreur minimale $\varepsilon_A(\boldsymbol{\vartheta})$ correspondante l'*erreur d'apprentissage*.

4.2.4 Validation

Dès le moment où on a obtenu un modèle, la question se pose de la validité de ce modèle. Nous voulons savoir si le modèle est adéquat pour l'application envisagée. Si nous prenons l'exemple d'un modèle d'un bassin versant, nous voulons savoir si on peut l'appliquer à un bassin spécifique ou à plusieurs bassins, si on peut l'utiliser dans toutes les situations météorologiques ou uniquement quand la précipitation ne dépasse pas un certain niveau.

Dans la phase de validation, on teste si le domaine de validité du modèle recouvre le domaine d'application envisagé. Sinon, il faut modifier le modèle ou limiter le domaine d'application.

Pour valider un modèle, on prend un critère d'erreur $\varepsilon(\boldsymbol{\vartheta})$, un *ensemble de validation* Y_V et l'ensemble d'estimations $\hat{Y}_V(\boldsymbol{\vartheta})$ correspondant. $\boldsymbol{\vartheta}$ était déjà fixé dans la phase d'apprentissage et on s'intéresse à l'*erreur de validation* ε_V qui correspond à un ensemble Y_V spécifique. On dira que le modèle est valide quand ε_V est moins grand qu'un certain seuil qui dépend des exigences de l'utilisateur.

Il est clair qu'on va prendre un ensemble de validation qui «décrit» le domaine d'application. Pour le modèle d'un bassin versant, ont pourra prendre

un Y_V qui comprend des mesures de sorties d'un bassin spécifique ou de plusieurs bassins.

Bien que les phases d'apprentissage et de validation ont des buts bien différents, le premier ne peut dans beaucoup de cas pas être vu indépendamment du deuxième. Souvent le constructeur du modèle (qui fait l'apprentissage) est identique à l'utilisateur (qui fait la validation) ou connaît ses exigences. Dans ce cas, il est de bonne pratique de choisir des ensembles Y_A et Y_V disjoints (par exemple deux sous-ensembles disjoints de l'ensemble des mesures disponibles Y). Ceci pour garantir que l'erreur de validation est une bonne estimation de l'erreur pendant l'utilisation (la vraie erreur) du modèle dans la pratique, où les couples $(y_k, \hat{y}_k)$ avec $k \notin I_A$ ne sont pas nécessairement rencontrés pendant l'apprentissage ou ne sont pas rencontrés avec la même fréquence.

Parfois on veut inclure tout les données disponibles dans l'ensemble d'apprentissage (par exemple quand le nombre de données est limité). Dans la section suivante, nous parlerons d'une méthode qui permet de le faire en même temps que de donner une bonne estimation de la vraie erreur.

4.2.5 Le bootstrap

Il existe plusieurs méthodes pour obtenir une estimation ε_V de la vraie erreur à partir de l'erreur d'apprentissage ε_A et l'ensemble d'apprentissage Y_A.[2] Tout ces méthodes sont asymptotiquement équivalentes, c'est à dire, les estimations qu'elles donnent vont converger vers un ε_V commun quand le nombre d'échantillons augmente. Une première classe de méthodes (AIC, BIC,...) utilise ε_A, le nombre d'échantillons dans l'ensemble d'apprentissage et le nombre de paramètres l utilisés. Une deuxième classe de méthodes (cross validation, le bootstrap,...) utilise ε_A et l'ensemble d'apprentissage Y_A (qui est égale à l'ensemble des données disponible Y). Nous utiliserons la méthode basée sur le bootstrap, qui est applicable pour des modèles très divers, même quand le nombre de paramètres l n'est pas connu.

Le *bootstrap* est une méthode de simulation pour faire des inférences statistiques. Cette méthode consiste en la création de différents ensembles Y_i d'échantillons à partir d'un ensemble initial Y. Chacun de ces ensembles Y_i est créé en faisant un nombre de tirages (avec remplacement) égal au nombre n d'échantillons dans Y.[3] On crée de cette façon n_B ensembles de bootstrap Y_i. Nous illustrons cette méthode à la figure 4.2.

Les différents ensembles Y_i peuvent maintenant être vus comme des ensembles d'apprentissage. Pour chacun de ces ensembles, nous pouvons calculer le vecteur de paramètres optimal $\boldsymbol{\vartheta}_i^*$ et l'erreur d'apprentissage $\varepsilon_{A,i}$. Chaque

2. Pour un aperçu de ces méthodes, voir [19, *Ch.17*].
3. Dans une terminologie statistique, on dit qu'avec la méthode du bootstrap on remplace la population avec l'échantillon original Y et on remplace l'échantillon original par un échantillon de bootstrap Y_i.

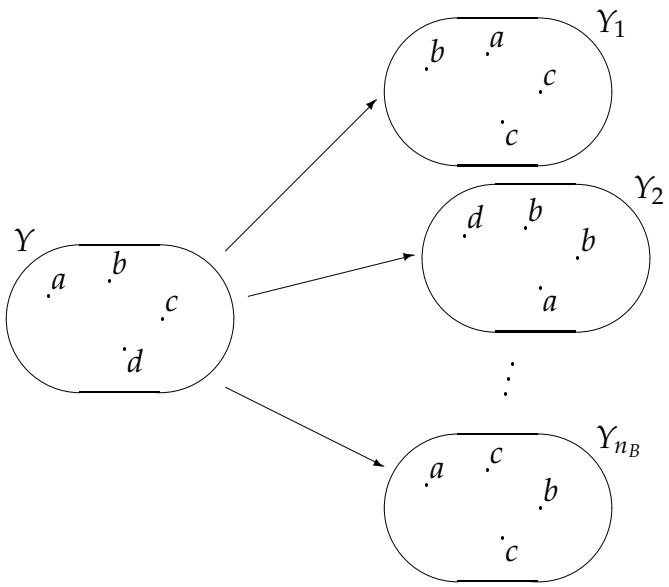

FIGURE 4.2: Illustration de la création des ensembles de bootstrap.

fois, nous pouvons utiliser l'ensemble initial Y comme ensemble de validation et calculer l'erreur de validation $\varepsilon_{V,i}$. L'estimation de la vraie erreur est calculée comme suit,

$$\varepsilon_V = \varepsilon_A + \frac{1}{n_B} \sum_i [\varepsilon_{V,i} - \varepsilon_{A,i}], \tag{4.3}$$

où ε_A est l'erreur d'apprentissage du modèle pour l'ensemble d'apprentissage Y.

L'équation (4.3) doit être interprétée comme une correction de biais de l'erreur ε_A. Le terme de correction, est une estimation de la différence entre l'erreur d'apprentissage et l'erreur de validation (appelé l'*optimisme*). Cette estimation est calculée comme une moyenne des différences entre l'erreur de validation et l'erreur d'apprentissage pour les ensembles de bootstrap Y_i. On obtient une meilleure estimation de l'optimisme en prenant un nombre élevé d'ensembles de bootstrap n_B.

Notons qu'il existe encore d'autres méthodes d'estimation de la vraie erreur basées sur la méthode du bootstrap, pour une description de ces méthodes et pour une dérivation de (4.3), voir [19, *§17.6.2, §17.7*].

L'estimation faite avec (4.3) est légèrement biaisée vers le bas, mais a une plus petite variance[4] que la méthode de cross validation (voir [19, *§17.8*]).

4. Les différentes estimations d'un même modèles varient parce chaque fois on utilise des différentes ensembles de bootstrap.

4.3 Modèles de régression

4.3.1 Le régresseur

Commençons en introduisant un peu de terminologie. Dans le cadre des modèles de régression on appellera les entrées $\boldsymbol{u}_{k+1}$ le *régresseur* et on le notera $\boldsymbol{G}_{k+1}$ pour souligner que ce vecteur est constitué principalement de mesures de grandeurs (obtenu éventuellement après un prétraitement) venant du système hydrologique.

Comme les régresseurs qu'on utilisera ne contiendront que des données disponibles avant l'instant $k+1$ pour estimer $\hat{y}_{k+1}$ on parlera dans la suite de la prédiction $\hat{y}_{k+1}$.

La prédiction avec les modèles de régression décrit ci-dessous s'écrit de façon générale comme

$$\hat{y}_{k+1}(\boldsymbol{\vartheta}) = \langle \boldsymbol{\vartheta}, \boldsymbol{G}_{k+1} \rangle,$$

où $\langle \cdot, \cdot \rangle$ désigne un produit scalaire. Dans la suite on ne notera plus explicitement la dépendance de la prédiction $\hat{y}_{k+1}$ des paramètres.

Quand on a choisi le modèle de régression et le critère d'erreur, il reste souvent une liberté de sélection du nombre d'éléments dans le régresseur et leur nature. Notons l la longueur du régresseur. Dans la plupart des cas quand l est petite, le modèle n'arrive pas à capter toute la dynamique de la système et la prédiction ne sera pas précise (c'est à dire, l'erreur de validation ε_V est trop élevé selon les besoins de l'utilisateur). Si, à l'autre extrême, l est pris trop grand, le modèle capte aussi le bruit contenu dans l'ensemble d'apprentissage et l'erreur d'apprentissage ε_A sera bas, mais l'erreur de validation remontera (avec un l croissant). Ce phénomène est appelé *surapprentissage*.[5] Il faut alors trouver le juste milieu entre les deux extrêmes. Ceci est fait en comparant ε_V pour des différentes longueurs l. Une considération qui doit être pris en compte est que le modèle doit être parcimonieux, c'est à dire, limiter la longueur l. Ceci est utile pour limiter les besoins (temps et technologie) liés au calcul associé à l'utilisation du modèle.

La sélection de la longueur du régresseur est illustré dans la figure 4.3 sur une courbe obtenu d'une simulation.

4.3.2 Régression linéaire : le modèle ARX

On parle d'un modèle d'un système comme étant une *régression linéaire* quand l'estimation $\hat{y}_{k+1}$ est une combinaison linéaire de mesures de certaines grandeurs du système.

5. Un exemple peut illustrer ce qui est le surapprentissage. Si les points qu'on veut prédire se trouve sur une droite et les grandeurs qu'on peut mettre dans le régresseur sont des versions bruitées des points précédents on veut bien sûr éviter d'aussi suivre l'évolution semblable du bruit.

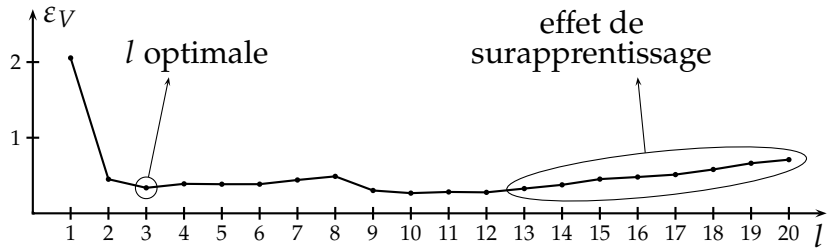

FIGURE 4.3: Illustration de la sélection de la longueur du régresseur.

Une régression linéaire est *auto-régressive* quand le régresseur contient des mesures passées de la sortie du système.

On parle d'un modèle auto-régressif (AR) quand ces mesures passées sont les seules valeurs dans le régresseur. Dans le cas où on considère les l dernières valeurs, ceci veut dire que

$$\boldsymbol{G}_{k+1} = (y_{k-l_y+1}, \ldots, y_k).$$

Alors on a que[6]

$$\hat{y}_{k+1} = \langle \boldsymbol{\vartheta}, \boldsymbol{G}_{k+1} \rangle = \boldsymbol{\vartheta} \cdot \boldsymbol{G}_{k+1} = \sum_{i=1-l_y}^{0} a_i y_{k+i},$$

où $\boldsymbol{\vartheta} = (a_{1-l_y}, \ldots, a_0) \in \mathbb{R}^{l_y}$.

Une régression linéaire peut utiliser d'autres valeurs que seulement des valeurs passées de la sortie. On parle d'un modèle *auto-régressif avec des variables exogènes* (ARX) quand il y a des éléments dans le régresseur qui ne sont pas des sorties passées.[7] Alors, si on utilise des x pour désigner les variables exogènes,

$$\boldsymbol{G}_{k+1} = (y_{k-l_y+1}, \ldots, y_k, x_{k-l_x+1}, \ldots, x_k, x_\alpha, x_\beta, x_\gamma, \ldots).$$

Dans la notation nous montrons que les variables exogènes peuvent être des grandeurs qu'on peut indexer avec des indices de temps, mais que ceci n'est pas toujours le cas (indices α, β, γ). Ce choix de régresseur entraîne que

$$\hat{y}_{k+1} = \sum_{i=1-l_y}^{0} a_i y_{k+i} + \sum_{j=1-l_x}^{0} b_j x_{k+j} + b_\alpha x_\alpha + b_\beta x_\beta + b_\gamma x_\gamma + \ldots.$$

Ici,

$$\boldsymbol{\vartheta} = (a_{1-l_y}, \ldots, a_0, b_{1-l_x}, \ldots, b_0, b_\alpha, b_\beta, b_\gamma, \ldots).$$

6. Il est utile de noter que ce n'est pas nécessaire que tout les instants sont représentés. On peut considérer une $I \subset \mathbb{Z}$ et écrire $\hat{y}_{k+1} = \sum_{i \in I} a_i y_{k+i}$, où bien sûr $\max_i I < 1$.
7. On ne considère pas le cas où il y a des mesures de sorties futures dans le régresseur.

Souvent on prend 1 comme une des variables, car dans la pratique un terme constant amène à une amélioration très importante (c'est à dire on remarque une réduction importante du minimum du critère d'erreur).

On peut se demander comment on doit faire le choix de la longueur l_x des différentes variables exogènes. Une méthode consiste d'utiliser la méthode illustré dans la figure 4.3 pour chaque variable, en gardant la partie du régresseur déjà déterminé fixé.

4.3.3 Régression non linéaire : le réseau RBF

On parle d'un modèle d'un système comme étant une *régression non linéaire* quand l'estimation $\hat{y}_{k+1}$ est une combinaison non linéaire de mesures de certaines grandeurs du système.

On note $\boldsymbol{G}_{k+1}$ de nouveau le régresseur, l'estimation est

$$\hat{y}_{k+1} = \langle \boldsymbol{\vartheta}, \boldsymbol{G}_{k+1} \rangle = \hat{g}_{\boldsymbol{\vartheta}}(\boldsymbol{G}_{k+1}),$$

où $g_{\boldsymbol{\vartheta}}$ est une fonction réelle paramétrée par $\boldsymbol{\vartheta}$.

Le cas de la régression linéaire est inclus pour $\hat{g}_{\boldsymbol{\vartheta}}$ égale à une combinaison linéaire des composantes de $\boldsymbol{G}_{k+1}$.

Quand on utilise un modèle non linéaire, intrinsèquement plus compliqué qu'un modèle linéaire comme l'ARX, il faut faire attention que les avantages (une meilleure estimation) contre-balancent les désavantages (des calculs plus lourds, l'augmentation du nombre de paramètres).

Pour faire un modèle de régression non linéaire d'un système, on peut utiliser le réseau à fonctions radiales de base qu'on a vu dans §3.4. Dans ce cas, l'estimation est une combinaison linéaire de fonctions radiales φ_i. Ce qui donne comme estimateur,

$$\hat{y}_{k+1} = \sum_{i=1}^{m} \lambda_i \varphi_i(\|\boldsymbol{G}_{k+1} - \boldsymbol{c}_i\|),$$

avec $\boldsymbol{c}_i$ un vecteur dans l'espace de régresseurs possibles appelé *centroïde*, m le nombre de centroïdes, $\lambda_i \in \mathbb{R}$ un poids et $\|\cdot\|$ une norme quelconque.

Le vecteur des paramètres (dans le cas d'un φ_i gaussienne d'écart type σ_i) est donc

$$\boldsymbol{\vartheta} = (\lambda_1, \ldots, \lambda_m, \sigma_1, \ldots, \sigma_m, c_{1,1}, \ldots, c_{m,l}).$$

Comme nous l'avons montré dans §3.4.2, les σ_i et les $\boldsymbol{c}_i$ ne sont pas choisis par minimisation de l'erreur. Dans le vecteur $\boldsymbol{\vartheta}$ on écrit uniquement les paramètres qui sont relevant pour l'estimation. Il se peut qu'on a utilisé des paramètres additionnels pendant l'apprentissage (par exemple κ et σ_{V_i} dans §3.4.2).

On voit que le nombre de paramètres utilisés avec ce modèle de régression est normalement beaucoup plus grand que pour un modèle ARX. Notamment, on y a un nombre de paramètres égal au nombre d'éléments l du régresseur

G_{k+1}. Pour un modèle de régression basé sur un réseau RBF avec m centroïdes et avec des fonctions de base φ_i à p paramètres on à $m + ml + mp$ paramètres en totale pour le modèle de régression.

4.4 Bibliographie

La plupart de l'exposé dans §4.2 est basée sur [20], un livre qui traite les différentes aspects de la modélisation de systèmes dynamiques. On a utilisé

[20, *Ch.9*] pour §4.2, §4.2.2 et §4.2.3 et
[20, *§1.5*] pour §4.2.4.

Ces section précédentes ont aussi été influencé par [2]. Pour la section §4.2.5 on a consulté [19, *Ch.1, Ch.17*].

Aussi pour §4.3.1 et §4.3.2 on a consulté [20, *Ch.9*]. La section §4.3.3 est inspiré par [15].

Le chapitre entier a pu bénéficier de l'expérience de A. Lendasse.

Chapitre 5

Méthodes de prédiction de débit

5.1 Introduction

Dans ce chapitre, deux méthodes de prédiction de débit seront décrites. La première, le modèle Hydromax, est un modèle qui utilise quelques connaissances théoriques sur les bassins versants. La deuxième est celle qui est développée dans le cadre de ce mémoire. Ce chapitre sera consacrée à la partie théorique des méthodes de prédiction, dans le chapitre 6 on prendra un point de vue plus pratique.

5.2 Hydromax

Hydromax est un modèle de prédiction de débit qui prend comme entrées des mesures de débit $Q(t)$, des mesures pluviométriques $P_i(t)$ et des données d'évapotranspiration $ETP(t)$. Dans une première phase la pluie brute $PB(t)$ est calculée avec la méthode de krigeage qu'on a vue dans §2.4. Les autres phases dans la prédiction seront décrites ci-dessous, où on utilisera la notation Q_k, PB_k et ETP_k introduit dans §2.5.

On considère la situation où on a des mesures jusqu'à $t = t_0$. Alors les données sont indexées par un k négatif et les prédictions par un k strictement positif.

5.2.1 Fonction de production

D'abord, la pluie nette PN_k, la partie de la pluie brute PB_k qui arrive à la rivière, est calculée. Ceci est fait en utilisant un modèle non linéaire, appelé *fonction de production*, qui décrit l'équilibre des volumes d'eau dans le bassin versant (l'eau dans le sol, l'eau dans la végétation, l'eau superficielle,...).

A chaque instant $t_0 + k\Delta t$ la pluie brute est décomposée en trois parties,

$$PB_k = PN_k + E_k^1 + W_k.$$

On a la pluie nette PN_k, le volume W_k d'eau qui sera stocké dans le bassin et un terme

$$E_k^1 = \min\{PB_k, ETP_k\}$$

qui représente l'évaporation immédiate. Ici, ETP_k désigne une estimation de l'évapotranspiration saisonnière potentielle. Par exemple, si k est un instant en hiver, alors $ETP_k = ETP_{\text{hivernale}}$.

Le stockage d'eau dans le bassin est représenté par un réservoir linéaire avec entrée W_k, décrit par l'équation aux différences

$$S_k = S_{k-1} + W_k - E_k^2 - I_k.$$

Ici S_k est le stockage d'eau dans le bassin à l'instant k,

$$I_k = \alpha\,[S_{k-1} + W_k]$$

est le volume perdu par percolation, représenté par une fonction linéaire du stockage d'eau disponible (α est le paramètre de percolation spécifique) et

$$E_k^2 = \max\{0, \min\{ETP_k - PB_k, S_{k-1} + W_k - I_k\}\}$$

représente le volume d'eau perdu par évapotranspiration. Il est supposé que le bassin ne peut que contenir un volume maximal $S_{\max}$.

Alors, le volume W_k d'eau qui sera stocké est exprimé en fonction de S_k et PB_k de manière que deux conditions soient satisfaites, c'est à dire

(i) $0 \leq S_k \leq S_{\max}, \quad \forall k$,

(ii) le principe hydrologique que PN_k est une fonction croissante de PB_k et S_k.

Dans ce cas, une fonction valable est

$$W_k = [S_{\max} - S_k]\left[1 - \exp(-\beta\frac{PB_k - E_k^1}{S_{\max} - S_k})\right].$$

On constate que ce modèle non linéaire utilise les trois paramètres α, β et $S_{\max}$ qui doivent être calibrés avec des données expérimentales pour chaque bassin considéré.

5.2.2 Prédiction à court terme

Un modèle de régression linéaire (un modèle ARX, voir §4.3.2) est appliqué. Ce modèle décrit le ruissellement de la pluie nette vers l'exutoire du bassin et calcule la prédiction du débit $\hat{Q}_h$. L'*horizon de prédiction* est donc $h\Delta t$. On utilise par conséquent une combinaison linéaire de termes de débit passés et de termes de pluie nette passés,

$$\hat{Q}_h = \sum_{k=1-n}^{0} a_k Q_{kh} + \sum_{l=1-m}^{0} b_l PN_{lh},$$

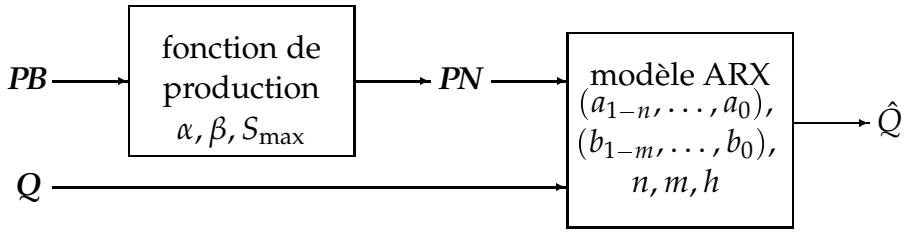

FIGURE 5.1: Schéma du modèle Hydromax.

où on voit que $h\Delta t$ est le temps entre les différentes mesures utilisées.[1]

Pour chaque bassin, les valeurs a_k, n, b_l, m et h sont déterminées à partir des données expérimentales. L'ensemble d'apprentissage utilisé pour l'identification est différent de l'ensemble de validation. $h\Delta t$ doit être plus petit que le temps de réponse naturel du bassin.[2] Les dimensions n et m sont sélectionnées par des méthodes standards de la théorie d'identification de systèmes.[3] Les coefficients a_k, b_l sont calibrés par régression linéaire.

5.2.3 Prédiction à long terme

Pour prédire des valeurs de débit sur des horizons qui sont significativement plus grands que le temps de réponse naturel du bassin de rivière, il nous faut bien sûr avoir des données sur de futures pluies nettes PN_k. Celles-ci peuvent être estimées (ce qui donne $\hat{PN}_k$) à partir des prévisions météorologiques PB_k. Ces prédictions à plus long terme sont obtenues du modèle à court terme par itérations successives :

$$\hat{Q}_{jh} = \sum_{k=j-n}^{0} a_{k-j+1} Q_{kh} + \sum_{k=1}^{j-1} a_{k-j+1} \hat{Q}_{kh} + \sum_{l=j-m}^{0} b_{l-j+1} PN_{lh} + \sum_{l=1}^{j-1} b_{l-j+1} \hat{PN}_{lh},$$

avec j strictement positif.

5.2.4 Structure de Hydromax

Après le calcul de la pluie brute PB_k le modèle utilisé par Hydromax est donc constitué de deux sous-modèles, un non linéaire (la fonction de production) et un linéaire (le modèle ARX utilisée pour la prédiction). Ceci est montré schématiquement dans la figure 5.1. On a indiqué les paramètres à détérminer.

---

1. Si par exemple $h = 6$, $n = 3$ et $m = 1$, on a que $\hat{Q}_6 = a_{-2}Q_{-12} + a_{-1}Q_{-6} + a_0 Q_0 + b_0 PN_0$. Si de plus $\Delta t = 1$h, la prédiction $\hat{Q}_6$ correspond à $\hat{Q}(t_0 + 6\text{h})$.
2. Il n'existe pas de définition stricte du terme «temps de réponse naturel», limitons-nous à dire que c'est une mesure de la réponse impulsionelle du bassin. Regardez [1] et [5, *III, Ch.1*] pour plus d'information. Dans la pratique, l'horizon de prévision $h\Delta t$ est une valeur entre 2h et 12h.
3. Regardez par exemple [20, *§10.4*].

Dans le cas de prédiction à court terme on a comme variables

$$\begin{aligned}\boldsymbol{PB} &= (PB_{[1-m]h}, \ldots, PB_0),\\ \boldsymbol{PN} &= (PN_{[1-m]h}, \ldots, PN_0),\\ \boldsymbol{Q} &= (Q_{[1-n]h}, \ldots, Q_0),\\ \hat{Q} &= \hat{Q}_h.\end{aligned}$$

Pour la prédiction à long terme on a

$$\begin{aligned}\boldsymbol{PB} &= (PB_{[j-m]h}, \ldots, PB_0, \hat{PB}_h, \ldots, \hat{PB}_{[j-1]h}),\\ \boldsymbol{PN} &= (PN_{[j-m]h}, \ldots, PN_0, \hat{PN}_h, \ldots, \hat{PN}_{[j-1]h}),\\ \boldsymbol{Q} &= (Q_{[j-n]h}, \ldots, Q_0, \hat{Q}_h, \ldots, \hat{Q}_{[j-1]h}),\\ \hat{Q} &= \hat{Q}_{jh}.\end{aligned}$$

5.3 Notre méthode

La méthode de prédiction de débit qu'on décrira ci-dessous prend comme entrées des mesures de débit $Q(t)$ et des mesures pluviométriques $P_i(t)$, mais elle ne prend pas de données d'évapotranspiration $ETP(t)$ (contrairement à Hydromax). Dans une première phase la pluie brute $PB(t)$ est calculée comme moyenne arithmétique des différentes mesures $P_i(t)$ (§2.4). Les autres phases de la prédiction seront décrites ci-dessous, où on utilisera la notation Q_k, PB_k.

De manière analogue à la section §5.2 on considère la situation où on a des mesures jusqu'à $t = t_0$. Alors les données sont indexées par un k négatif et les prédictions par un k strictement positif. Comme pour Hydromax, il faut construire un modèle différent pour chaque bassin.

5.3.1 Aperçu de la méthode de prédiction

Si avec Hydromax on faisait des prédictions successives de débits $\hat{Q}_{jh}, j > 0$ individuels, le but maintenant est de prédire une succession de débits $\hat{\boldsymbol{Q}}_0 = (\hat{Q}_h, \ldots, \hat{Q}_{nh})$.[4]

La série temporelle qu'on considère est donc constituée d'éléments $\boldsymbol{Q}_k = (Q_{k+h}, \ldots, Q_{k+nh})$ appelés *courbes de débit*.[5] La prédiction se fait en plusieurs étapes :

1. D'abord les courbes de débit $\boldsymbol{Q}_k$ sont divisées en trois parties. La première est sa moyenne
$$\mu_k = \frac{1}{n}\sum_j Q_{k+jh},$$

4. L'*intervalle de temps* h pour notre méthode n'est pas nécessairement égale à le h pour Hydromax.
5. Rappelons que pour les données disponibles nous avons $k + nh \leq 0$.

la deuxième est son écart type

$$\sigma_k = \sqrt{\frac{1}{n}\sum_j [Q_{k+jh} - \mu_k]^2},$$

et la dernière est *le profil*

$$\boldsymbol{\pi}_k = \frac{(Q_{k+h} - \mu_k, \ldots, Q_{k+nh} - \mu_k)}{\sigma_k}.$$

Dans cette étape, nous avons en fait créé trois séries temporelles distinctes.[6]

2. Puis, on fait des prédictions pour les trois parties sur base des moyennes, écarts types et profils passés, ce qui donne $\hat{\mu}_0$, $\hat{\sigma}_0$ et $\hat{\boldsymbol{\pi}}_0$. Dans des sections suivantes on examinera dans plus de détail la façon dont on prédit ces quantités.
3. Finalement, la prédiction de la courbe de débit est calculé,

$$\hat{\boldsymbol{Q}}_0 = \hat{\sigma}_0 \hat{\boldsymbol{\pi}}_0 + \hat{\mu}_0 \mathbf{1},$$

où on a utilisé la notation $\mathbf{1} = (1, \ldots, 1)$.

Dans la figure 5.2 nous avons schématiquement montré les étapes dans la méthode de prédiction. A gauche on montre la création des trois séries temporelles et à droite on montre la combinaison des prédictions pour ces trois séries temporelles.

Dans la littérature des descriptions et des applications de cette méthode peuvent être trouvé dans [11], [13], [14] et [2]. L'application type dans ces articles est la prédiction de la consommation journalière d'électricité, c'est à dire, on veut prédire des courbes $\boldsymbol{C}_0 = (C_1, \ldots, C_{24})$ où C_j est une quantification de la consommation horaire d'électricité.

La grande différence entre la situation rencontrée dans la littérature et la situation dans laquelle nous nous trouvons est le caractère des données. La consommation d'électricité à un caractère journalier lié au rythme journalier de notre société. Le débit d'une rivière ne connaît pas un tel caractère, sauf les variations saisonnières, mais celles-ci se manifestent sur des plages de temps beaucoup plus grandes que celles considérées dans la prédiction. Cette différence se reflète dans la façon de tirer des courbes des mesures disponibles. Pour la consommation il était déjà clair qu'on prend les courbes correspondantes au jour. Pour le débit d'une rivière (si $h = 1$ et $n = 24$) nous prenons toutes les courbes de 24 mesures successives. Les deux différentes situations sont illustrées dans la figure 5.3. Là-dedans on a utilisé la notation

$$\boldsymbol{C}_k = (C_{24k+h}, \ldots, C_{24k+nh}).$$

6. Cette division est faite pour pouvoir faire une classification des profils avec une carte auto-organisatrice, voir §5.3.3.1.

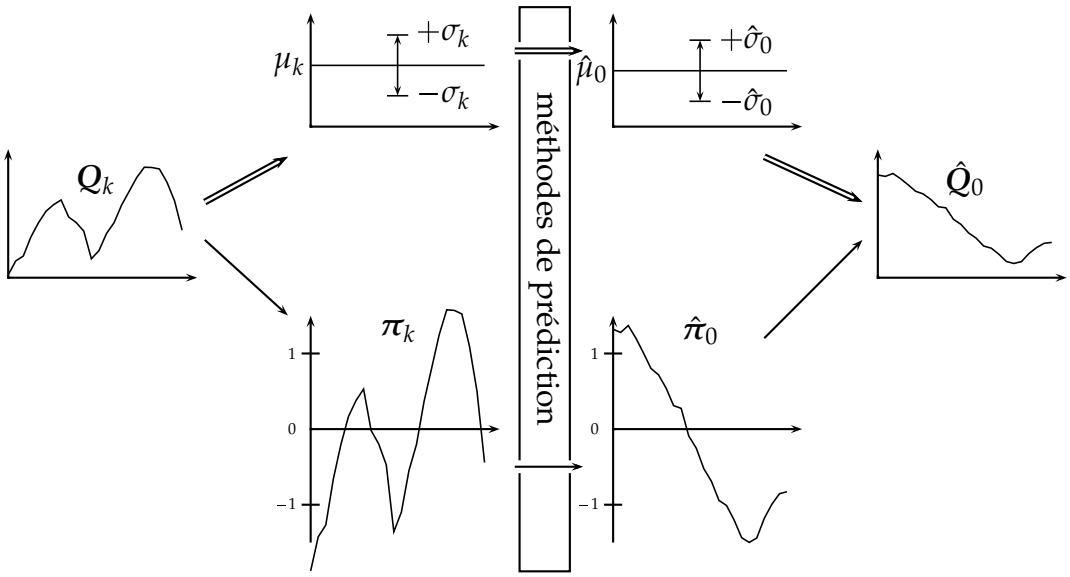

FIGURE 5.2: Schéma des étapes de la méthode de prédiction.

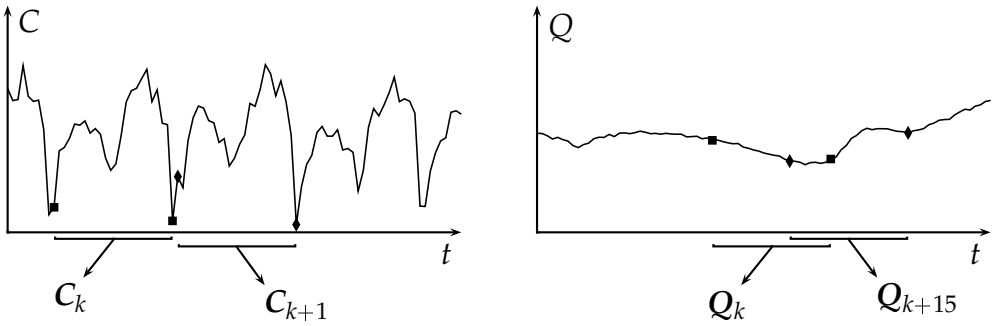

FIGURE 5.3: Construction des courbes de consommation d'électricité et des courbes de débit.

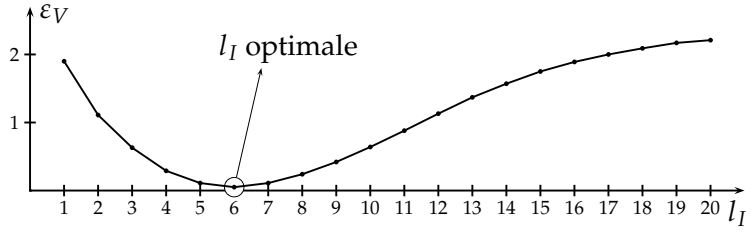

FIGURE 5.4: Illustration de la sélection du nombre d'éléments dans I.

5.3.2 Prédiction de la moyenne et de l'écart type

Les prédictions de la moyenne μ et de l'écart type σ peuvent être obtenues par des modèles de régression classique. Ceci peut être soit un modèle linéaire, soit un modèle non linéaire comme nous les avons vus dans §4.3.

Comme on l'a vu dans §4.3.1, on peut alors calculer les prédictions comme suit,

$$\hat{\mu}_0 = \langle \boldsymbol{\vartheta}_\mu, \boldsymbol{G}_{\mu,0} \rangle,$$
$$\hat{\sigma}_0 = \langle \boldsymbol{\vartheta}_\sigma, \boldsymbol{G}_{\sigma,0} \rangle.$$

Les paramètres $\boldsymbol{\vartheta}_\mu$ et $\boldsymbol{\vartheta}_\sigma$ sont déterminés comme décrit dans §4.2.3.

Dans les régresseurs $\boldsymbol{G}_{\mu,0}$ et $\boldsymbol{G}_{\sigma,0}$ on peut mettre toute une panoplie de différentes grandeurs. Des choix évidents (mais pas nécessaires) sont les variables «endogènes» μ_k et σ_k. Comme variables exogènes on a le choix parmi

- une constante (c'est à dire 1),
- respectivement σ_k et μ_k ($k \leq 0$),
- des débits passés Q_k ($k \leq 0$),
- des pluies passées PB_k ($k \leq 0$),
- des prévisions météorologiques PB_k ($k > 0$),
- une somme de pluies passées $\sum_{k \in I} PB_k$ ($\max I \leq 0$),
- une somme de prévisions météorologiques $\sum_{k \in I} PB_k$ ($\min I > 0$),
- une somme de pluies passées et de prévisions météorologiques $\sum_{k \in I} PB_k$.

Pour chacune de ces variables possibles, il faux choisir combien on en prend et lesquelles (pour quels instants, déterminés par les k et les I). Il est clair qu'il y a un nombre énorme de régresseurs possibles, qu'on peut déjà fort réduire en se basant sur l'expérience et un peu de bon sens.

Pour choisir combien de variables d'un même type sont à inclure, on peut utiliser la méthode décrite dans la section §4.3.1. Pour les variables qui sont des sommes, il faut déterminer le nombre d'éléments l_I dans I. L'interprétation de la courbe correspondante est différente de celle illustrée dans la figure 4.3, car maintenant il ne s'agit chaque fois que d'une variable. Dans ce cas il faut juste choisir un l_I correspondant à un minimum de l'erreur de validation ε_V, comme illustré dans la figure 5.4.

5.3.3 Prédiction du profil

La prédiction du profil $\boldsymbol{\pi}_0$ est plus compliquée que la prédiction de la moyenne ou l'écart type. Il serait possible d'utiliser un modèle de régression linéaire ou non linéaire itérativement ou d'utiliser n de ces modèles (un par composante $\pi_{0,j}$ de $\boldsymbol{\pi}_0$), ce qu'on appelle une méthode vectorielle. Nous avons cependant choisi d'examiner les possibilités de généraliser la méthode de la prédiction du profil décrite dans [11], [13], [14] et [2].[7]

5.3.3.1 Classification non supervisée

A la base de la méthode est une *classification non supervisée* des différents profils $\boldsymbol{\pi}_k$. Cette classification est faite avec une carte auto-organisatrice.

Nous allons utiliser les cartes auto-organisatrices à produit scalaire définies dans §3.3.5. Dans §A.1 nous montrons pourquoi nous avons fait ce choix. Nous y expliquons aussi qu'on ne doit pas faire de prétraitement des courbes de débit.

5.3.3.2 Méthodes de prédiction pour le profil dans la littérature

Pour introduire la méthode de prédiction du profil de $\boldsymbol{Q}_0$, il est utile d'examiner les variantes de la méthode de prédiction du profil de $\boldsymbol{C}_0$ rapporté dans la littérature.

Dans [13] et [11] on a construit une carte auto-organisatrice (rectangulaire, avec centroïdes $\boldsymbol{c}_i$) sur base des données pour plusieurs années. La carte est ensuite utilisée pour identifier un certain nombre de *types de jours*. Les distinctions qu'on peut faire sont la différence entre les jours de la semaine et le dimanche, les jours des différentes saisons, etc. Ceci est possible grâce au fait que les centroïdes similaires se trouvent proches l'un de l'autre dans la carte. On peut donc identifier des groupes de classes dans la carte qui correspondent à les différentes types de jours. Après qu'on a fait ce regroupement, on fait une matrice contenant le nombre de fois $a_{l,i}$ qu'un profil d'un jour du type l appartient à une zone de Voronoi V_i (ou la classe i) associée à un centroïde $\boldsymbol{c}_i$. Comme on connaît le type du jour, disons l, pour lequel on cherche à prédire $\boldsymbol{C}_0$, la prédiction du profil est définie d'être

$$\hat{\boldsymbol{\pi}}_0 = \boldsymbol{\pi}_l = \frac{\sum_i a_{l,i}\boldsymbol{c}_i}{\sum_i a_{l,i}}.$$

Il est clair qu'on ne peut pas utiliser un analogue du type de jour pour prédire les profils des courbes de débit parce que le débit d'une rivière n'a pas de dépendances temporelles pareilles. Cependant, cette méthode de prédiction

7. Cette méthode permet d'éviter quelques désavantages des méthodes itératives et vectorielles (regardez [13] et [11]).

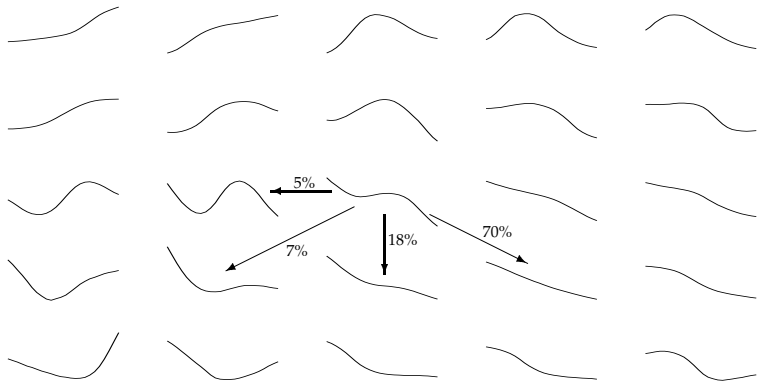

FIGURE 5.5: Des probabilités de transition dans une carte auto-organisatrice.

nous montre que les cartes auto-organisatrices peuvent être utilisées pour faire du regroupement.

Dans [14] et [2] un même type de carte que dans le paragraphe précédent est utilisé. Maintenant, on calcule les probabilités de transition entre les différentes zones de Voronoi V_j et $V_{j'}$. On considère deux profils $\boldsymbol{\pi}_k$ et $\boldsymbol{\pi}_{k+1}$ correspondant avec deux courbes de consommation d'électricité successives $\boldsymbol{C}_k$ et $\boldsymbol{C}_{k+1}$. Ces profils appartiennent respectivement à V_j et $V_{j'}$. La probabilité de transition est alors calculé comme suit,

$$\begin{aligned}\mathcal{P}(\boldsymbol{\pi}_k \in V_j \ \wedge \ \boldsymbol{\pi}_{k+1} \in V_{j'}) &= \frac{\mathcal{P}(\boldsymbol{\pi}_k \in V_j \mid \boldsymbol{\pi}_{k+1} \in V_{j'})}{\mathcal{P}(\boldsymbol{\pi}_k \in V_j)} \\ &= \frac{|\{\boldsymbol{\pi}_k \in V_j \mid \boldsymbol{\pi}_{k+1} \in V_{j'}\}|}{|\{\boldsymbol{\pi}_k \in V_j\}|},\end{aligned}$$

où $|X|$, avec X un ensemble, désigne la cardinalité de X. Dans la figure 5.5 nous donnons une illustration des probabilités de transition dans une carte auto-organisatrice. Dans [14] la prédiction du profil (si on connaît la zone de Voronoi V_j à laquelle appartient le profil actuel $\boldsymbol{\pi}_{-1}$) devient

$$\hat{\boldsymbol{\pi}}_0 = \sum_{j'} \mathcal{P}(\boldsymbol{\pi}_k \in V_j \ \wedge \ \boldsymbol{\pi}_{k+1} \in V_{j'}) \boldsymbol{c}_{j'}. \tag{5.1}$$

Cette méthode, qui prend une moyenne pondérée des centroïdes, peut être appliqué quand les centroïdes qui ont un poids non négligeable se trouvent proche l'un de l'autre dans la structure (pour qu'ils soient assez similaire). Dans [2], on prend comme prédiction le centroïde qui est associé à la plus grande probabilité de transition,

$$\hat{\boldsymbol{\pi}}_0 = \boldsymbol{c}_{\text{argmax}_{j'} \mathcal{P}(\boldsymbol{\pi}_k \in V_j \wedge \boldsymbol{\pi}_{k+1} \in V_{j'})}. \tag{5.2}$$

Pour cette méthode, il faut éviter des situations où un trop grand nombre de transitions a une probabilité non négligeable, certainement si les centroïdes

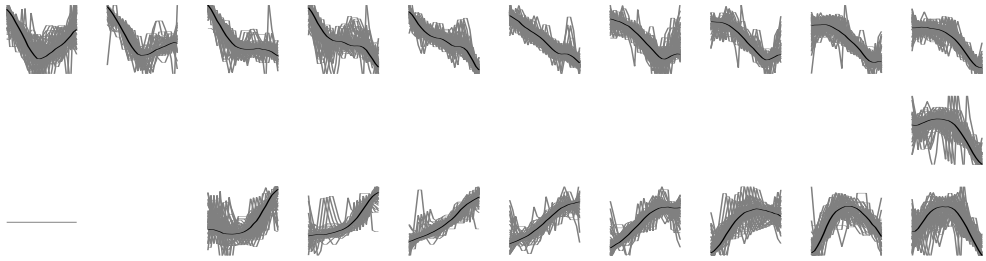

FIGURE 5.6: Visualisation de la variation dans chacune des classes d'une ficelle.

correspondants sont trop dissimilaires. Les méthodes décrites dans ce paragraphe nous ont montré que le profil précédent peut nous aider à faire une prédiction de profil.

5.3.3.3 Notre méthode pour la prédiction du profil

Maintenant, nous sommes assez préparés pour examiner la méthode de prédiction que nous proposons. La description dans cette section sera générale, c'est à dire, nous esquisserons les étapes nécessaires pour pouvoir faire des prédictions. L'application de cette méthode sera faite dans le chapitre 6.

1. D'abord, il faut construire une carte auto-organisatrice. Même si on pouvait a priori choisir une structure quelconque, il deviendra clair[8] qu'une structure unidimensionnelle (c'est à dire, une ficelle) est la plus commode.
   La taille de la carte (c'est à dire, le nombre de centroïdes) doit être choisie telle que la variation dans une classe est assez petite. La *variation d'une classe* correspondante à un centroïde $\boldsymbol{c}_i$ peut par exemple être mesurée par l'écart type des profils se trouvant dans la zone de Voronoi correspondante V_i. La variation peut aussi être visualisée en dessinant tous les profils appartenant à une classe dans le même graphique que le centroïde de la classe (regardez la figure 5.6).
2. La variation dans presque toutes les classes reste trop grande pour être utilisable pour la prédiction, même avec des cartes de très grande taille (avec un centaine de centroïdes).
   Comme deuxième étape on doit faire une analyse des courbes de débit afin d'arriver à un certain nombre de *situations* pour lesquelles des procédures de prédictions différentes seront utilisées. Ces situations sont idéalement choisies avec des «situations du terrain» en tête. Un exemple simpliste d'une telle situation est de considérer tous les profils $\boldsymbol{\pi}_k$ pour lesquels la dernière mesure de débit Q_k dépasse le seuil de pré-alerte c_{PA}.

8. Dans §6.2.2, la section où on applique la méthode de prédiction du profil décrit ici.

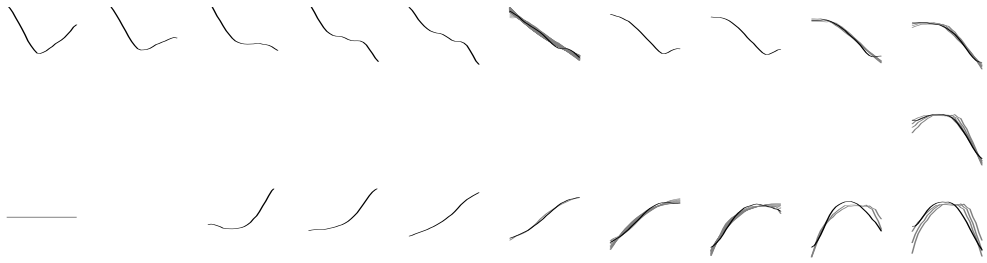

FIGURE 5.7: Visualisation d'une situation et la réduction correspondante des variations.

Avec chaque situation il ne correspond qu'un sous-ensemble des profils trouvé dans l'ensemble d'apprentissage. Comme on garde la carte auto-organisatrice déjà faite, ceci implique que le nombre de profils qui se trouve dans les classes est fort réduit. On «supprime» les profils qui ne correspondent pas à la situation étudiée.
Pour chaque situation on peut alors examiner la variation dans une classe. Une situation peut être considérée bonne quand la variation dans chacune des classes est petite. Pour illustrer ce qu'on vient d'expliquer, nous montrons dans la figure 5.7 la carte de la figure 5.6, mais maintenant pour la situation simpliste de notre exemple. Remarquez la réduction des variations.

3. Au moment qu'on se trouve dans une certaine situation on a à disposition une classification des profils correspondants à cette situation (voir la figure 5.7).
   Ces profils π_k seront utilisés pour construire une *procédure de prédiction de profil* pour la situation concernée. Dans cette procédure on peut tenir compte des profils précédents π_{k+nh}, comme on utilisait des probabilités de transition dans [14] et [2]. Nous ne prendrons ni (5.1), ni (5.2) comme prédiction, mais ferons un choix entre les transitions qui ont une probabilité non négligeable à l'aide des variables exogènes vues dans §5.3.2.
   Donnons un exemple simpliste pour éclaircir. Supposons que la classe de π_{-24} est 18.[9] Dans l'ensemble d'apprentissage les transitions à partir de 18 avec une probabilité non négligeable sont $18 \to 15$ et $18 \to 12$. On détermine que dans l'ensemble d'apprentissage la première transition correspond à une dernière mesure de pluie brute qui est au dessus de 4mm, et que la deuxième transition correspond à une dernière mesure de pluie brute qui est en dessous de 4mm. Sur base de PB_0 on peut alors déterminer la classe du profil π_0.

9. Les classes de la carte de la figure 5.7 peuvent être numérotées de 1 à 19 en partant en haut à gauche avec 1 et puis de suivant la ficelle.

Cette recette pour une méthode de prédiction du profil sera appliquée, et ainsi éclaircie, dans le chapitre 6.

On pourrait immédiatement se demander pourquoi on ne fait pas d'abord la sélection des situations et ensuite une classification non supervisée pour chaque situation. Il est vrai que cette approche serait possible, mais pour certaines situations, il se peut qu'il n'y a pas assez de profils pour construire la carte auto-organisatrice. L'avantage principal de cette approche serait d'avoir des centroïdes (utilisés comme prédictions) qui sont plus similaires aux profils dans leurs zones de Voronoi que les centroïdes d'une carte faite sur base d'un ensemble plus étendu. Cet effet peut cependant aussi être obtenu en prenant non pas les centroïdes $\boldsymbol{c}_j$ comme prédictions, mais la moyenne des profils se trouvant dans la zone de Voronoi V_j pour la situation,[10]

$$\hat{\boldsymbol{\pi}}_0 \propto \sum_{\substack{\boldsymbol{\pi}_k \in V_j \\ \boldsymbol{\pi}_k \sim \text{ situation}}} \boldsymbol{\pi}_k.$$

C'est cette approche qu'on prendra dans le chapitre 6.

Nous terminons cette section avec une remarque sur les caractérisations qualitatives (comme : variation assez petite, probabilités de transition négligeables) qu'on a parfois utilisées. Si les variations étaient assez petites ou si quelques probabilités de transition étaient négligeables dépend bien sûr de la qualité de la procédure de prédiction construite.

5.3.4 Structure de notre modèle

Comme nous l'avons fait pour Hydromax dans la figure 5.1, nous montrons dans la figure 5.8 le schéma de notre méthode. Contrairement à la figure 5.2, nous nous concentrons ici sur les modèles de prédiction.

Là-dedans, nous avons utilisé la notation suivante.

$$\begin{aligned}
\boldsymbol{Q} &= (\ldots, \boldsymbol{Q}_{-24}), \\
\boldsymbol{\mu} &= (\ldots, \boldsymbol{\mu}_{-24}), \\
\boldsymbol{\sigma} &= (\ldots, \boldsymbol{\sigma}_{-24}), \\
\boldsymbol{\pi} &= (\ldots, \boldsymbol{\pi}_{-24}), \\
\boldsymbol{i} &= (\ldots, i_{-24}),
\end{aligned}$$

où i_k est la classe du profil $\boldsymbol{\pi}_k$. Le nombre d'éléments dans chacun des vecteurs ci-dessus dépend des détails des trois modèles de prédiction.

10. Rappelons qu'on a enlevé un grand nombre de profils de l'ensemble d'apprentissage (ou équivalent : un grand nombre de points des zones de Voronoi) en considérant une situation quelconque. Ceci implique que les centroïdes ne correspondent plus nécessairement aux moyennes des profils dans les zones de Voronoi.

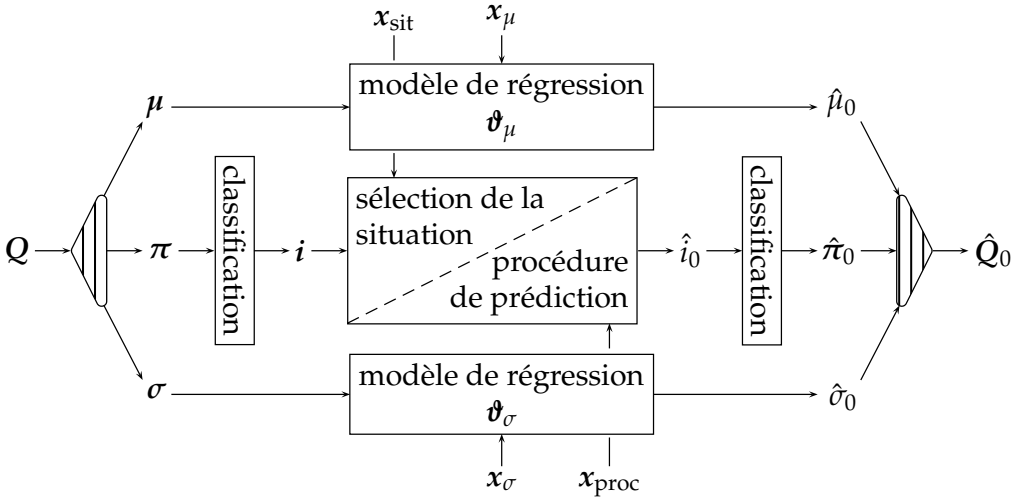

FIGURE 5.8: Schéma de notre méthode de prédiction.

Pour les modèles de régression (avec paramètres ϑ_μ et ϑ_σ), les régresseurs contiennent des éléments de μ et σ respectivement, mais aussi des variables exogènes contenus dans x_μ et x_σ. On a déjà parlé des variables exogènes possible dans §5.3.2.

Pour déterminer la situation on utilise les variables exogènes dans x_{sit} et éventuellement les classes i des profils passés. La procédure de prédiction du profil utilise i et les variables dans x_{proc}. Comme les prédictions sont faites séparément, on peut aussi utiliser $\hat{\mu}_0$ et $\hat{\sigma}_0$ dans x_{sit} et x_{proc}.

5.3.5 Choix à faire dans la méthode

Il doit être clair que notre méthode, comme nous l'avons décrite ci-dessus n'est pas encore applicable. Il y a pas mal de choix à faire.

Les premiers choses qu'on doit fixer sont la longueur des courbes de débit n et l'intervalle de temps h. Ces valeurs dépendent de la plage de prédiction nh voulue et de l'intervalle de temps optimal, à déterminer dans la pratique. Il est même possible d'utiliser des courbes de débit non régulières, c'est à dire, pour laquelle h n'est pas constant. Par exemple quand on veut des prédictions $\hat{Q}_0$ qui sont initialement plus détaillées.

Pour la prédiction de la moyenne et de l'écart type, on a un seul but : minimiser l'erreur de prédiction. On peut librement choisir le modèle de régression (ceux dont on a parlé, mais aussi d'autres) et les grandeurs dans le régresseur. Nous nous sommes pas concentrés sur cet aspect, mais il existe beaucoup de références sur le choix du régresseur (un point de départ est par exemple [20]).

Pour la prédiction du profil, il y a d'abord la taille de la carte[11] auto-organi-

11. Nous supposons ici qu'on utilise une ficelle. Si certaines conditions sont satisfaites, d'autres structures pourraient être considérées. On en parlera plus en détail dans §6.4.

satrice à choisir. Cette taille doit être choisie sur base de la qualité des procédures de prédiction qu'on peut en faire. Le deuxième choix est celui de la division en situations. La qualité des procédures de prédiction est de nouveau le critère. Les procédures de prédiction même et les grandeurs qu'on utilise là-dedans sont les derniers choix à faire. On en donne un exemple dans §6.2.2.3.

5.4 Comparaison

Examinons d'abord les données dont on a besoin pour les deux méthodes de prédiction. Toutes les deux utilisent des mesures de débits passées Q_k, des mesures de pluie P_k et des prévisions météorologiques PB_k. En plus, Hydromax utilise des données ETP_k sur l'évapotranspiration. Dans la forme que nous l'avons présentée, notre méthode de prédiction utilise uniquement les mesures de pluie pour le calcul de la pluie brute PB_k (voir §2.4). Si on utilisait le système de krigeage comme dans Hydromax, on aurait en plus besoin des positions z_j des pluviomètres et de la surface $|\Omega|$ du bassin versant.

Le degré de connaissances théoriques qui est utilisé dans les deux modèles diffère beaucoup. Notre méthode est basée sur les données : hors les interprétations hydrologiques liées aux données (qui sont utilisées pour calculer la pluie brute et de la définition éventuelle des situations), on n'utilise pas de connaissances hydrologiques. Hydromax par contre utilise une fonction de production (§5.2.1) qui décrit ce qui se passe avec la précipitation.

Quant aux résultats, il y a aussi des différences importantes. Hydromax donne une série de prédictions itératives. C'est à dire, la prédiction à court terme $\hat{Q}_h$ est entièrement basée sur des mesures, mais les prédictions à long terme $\hat{Q}_{jh}, j > 1$ sont de plus en plus basées sur des prédictions précédentes. Il s'en suit que l'erreur de prédiction attendue croîtra avec j. Avec notre méthode, on fait en une fois n prédictions $(\hat{Q}_h, \ldots, \hat{Q}_{nh}) = \hat{\boldsymbol{Q}}$. L'erreur attendue sera alors constante pour tout les $\hat{Q}_{jh}, 1 \leq j \leq n$.

5.5 Bibliographie

La section §5.2 est basée sur [1].

On s'est basé sur [11], [13], [14] et [2] pour écrire §5.3.1. Les mêmes références ont été utilisées dans §5.3.3. Elles ont aussi été utilisées comme inspiration pour notre méthode de prédiction de profil décrite dans la même section. Les références [12] et [21] ont été intéressantes pour bien comprendre comment on peut faire de l'analyse avec des cartes auto-organisatrices.

L'approche prise dans §5.4 est légèrement influencée par [4].

Chapitre 6

Application des méthodes de prédiction

6.1 Introduction

Ce chapitre contient la description et les résultats de l'application de notre méthode de prédiction. La méthode sera appliquée aux données décrites dans le chapitre 2. On se concentrera sur le bassin de la Lesse à Gendron.[1] Des trois années de données horaires disponibles, on a utilisé les deux premières années ('93 et '94) pour construire le modèle de prédiction et la dernière année ('95) pour le tester. Autrement dit, les mesures de '93 et de '94 fonctionnent comme ensemble d'apprentissage et celles de '95 comme ensemble de validation.

A la fin de ce chapitre, on examinera aussi les résultats obtenus avec Hydromax et on les comparera avec ceux obtenus par notre méthode. L'ensemble d'apprentissage utilisé pour l'identification du modèle Hydromax a une longueur d'une année.

6.2 Notre méthode

Nous avons appliqué notre méthode avec les choix suivants pour les paramètres des courbes de débit Q_k :

$$n = 24,$$
$$h = 1\text{h}.$$

Autrement dit, on fera des prédictions pour les vingt-quatre prochaines heures et dans ce choix nous suivons les articles [11], [13], [14], [14] et [2] sur la consommation d'électricité.

1. On a aussi appliqué des éléments de la méthode (prédiction de la moyenne et la construction d'une carte auto-organisatrice) aux données du bassin de l'Ourthe à Tabreux. Les résultats pour ces aspects étaient comparables avec ceux obtenus pour le bassin de la Lesse à Gendron.

6.2.1 Prédiction de la moyenne et de l'écart type

Pour la prédiction de μ_0 et σ_0 on a d'abord essayé d'utiliser un modèle ARX (voir §4.3.2). Avec ce modèle on a cherché les régresseurs (avec une longueur raisonnable) qui donnait des erreur ε_V minimales.

Pour la prédiction de la moyenne μ_0 nous trouvions un régresseur $\boldsymbol{G}_{\mu,0}$ contenant cinq composantes (le modèle contient donc 5 paramètres), dont

- une constante,
- les deux derniers mesures de débit Q_0 et Q_{-1},
- une somme de pluies passées $\sum_{k=-6}^{0} PB_k$ et
- une somme de prévisions météorologiques $\sum_{k=1}^{11} PB_k$.

L'erreur quadratique moyenne normalisée ($\varepsilon_{\overline{\text{QM}}}$, voir (4.2)) correspondante est $\varepsilon_V = 3{,}4\%$. Ceci semble un bon résultat, mais il faut faire attention. Comme l'écart type des moyennes est 33,3 m³/s, ceci correspond à une erreur quadratique moyenne de $[6{,}1\,\text{m}^3/\text{s}]^2$. Sachant que le débit moyen est 21,7 m³/s, l'erreur se révèle d'être pas négligeable. Quand on compare les prédictions avec les vrais valeurs, on voit que la différence entre les deux est surtout important pour les grands débits (les périodes de crues).

Pour la prédiction de l'écart type σ_0 nous trouvions un régresseur $\boldsymbol{G}_{\sigma,0}$ contenant dix composantes (le modèle contient donc 10 paramètres), dont

- une constante,
- les deux derniers mesures de débit Q_{-1} et Q_0,
- une somme de pluies passées $\sum_{k=-5}^{0} PB_k$,
- une somme de prévisions météorologiques $\sum_{k=1}^{13} PB_k$,
- trois écart types passés σ_{-24}, σ_{-25}, σ_{-26} et
- deux moyennes passées μ_{-24}, μ_{-25}.

L'erreur quadratique moyenne normalisée correspondante est $\varepsilon_V = 31{,}5\%$. L'écart types des écart types est 5,9 m³/s et l'erreur quadratique moyenne devient alors $[3{,}3\,\text{m}^3/\text{s}]^2$. Vue que la moyenne des écart types est 1,7 m³/s, cette erreur est clairement trop grand.

Il est devenu clair qu'on doit améliorer ces résultats pour que notre méthode de prédiction soit utilisable. Les améliorations qu'on pourra faire en modifiant le régresseur sont limitées. Nous avons trouvé qu'en ajoutant encore d'autres grandeurs, l'erreur ne diminue presque pas. On pourra éventuellement changer le temps d'écart (ici 1h) entre les débits Q_k, les moyennes μ_k et les écarts types σ_k inclus.

Pour encore obtenir une réduction de l'erreur de prédiction, il faut utiliser un modèle de régression non linéaire. Nous allons utiliser les réseaux à RBF (voir §4.3.3). Les régresseurs utilisés sont les mêmes que pour l'ARX, mais sans la composante constante.

Pour la prédiction de la moyenne μ_0 nous trouvions une erreur quadratique moyenne normalisée $\varepsilon_V = 0{,}7\%$. L'erreur quadratique moyenne devient alors

$[2{,}7\,\mathrm{m}^3/\mathrm{s}]^2$. On voit qu'on obtient une amélioration importante. Le réseau à RBF utilisé est constitué de 55 centroïdes ($\in \mathbb{R}^4$). Le nombre total de paramètres (voir §4.3.3) est alors 330. Nous avons aussi regardé les résultats si on n'utilise pas les prévisions météorologiques. Nous trouvions $\varepsilon_V = 2\%$ et $[4{,}7\,\mathrm{m}^3/\mathrm{s}]^2$ pour un réseau à 25 centroïdes (et donc 125 paramètres).

Pour la prédiction de l'écart type σ_0 nous trouvions une erreur quadratique moyenne normalisée $\varepsilon_V = 11{,}5\%$. L'erreur quadratique moyenne devient alors $[2\,\mathrm{m}^3/\mathrm{s}]^2$. Ici on obtient une amélioration encore plus importante. Le réseau à RBF utilisé est constitué de 150 centroïdes ($\in \mathbb{R}^9$). Le nombre total de paramètres (voir §4.3.3) est alors 1650.

Dû à la limitation du temps, nous n'avons pas pu appliquer les modèles décrit ci-dessus pour faire des prédictions de μ_0 et σ_0.

6.2.2 Prédiction du profil

Rappelons que la construction de la méthode de prédiction du profil consiste en trois étapes (voir §5.3.3.3),

1. Construction d'une carte auto-organisatrice.
2. Analyse des courbes de débit et définition des situations.
3. Construction d'une procédure de prédiction pour chaque situation.

6.2.2.1 Construction d'une carte auto-organisatrice

Pour la construction de la carte on a essayé deux types de structures, des cartes rectangulaires et des ficelles. Dans les figures 6.1 et 6.2 on montre des cartes obtenues.[2] On y a aussi montré tous les profils de l'ensemble d'apprentissage dans leurs propres classes et le nombre de profils par classe. On voit sur ces figures qu'ils ont la propriété qu'on attend d'une carte auto-organisatrice. Notamment, les centroïdes qui se trouvent proches l'un de l'autre sont similaires. On donne de l'information complémentaire sur la construction des cartes auto-organisatrices dans §A.1.

Dans la suite nous allons considérer le cas de la ficelle où chaque centroïde a deux plus proches voisins. Le cas d'une carte rectangulaire, ou chaque centroïde a quatre (ou huit) plus proches voisins, compliquerait l'étape trois, la construction d'une procédure de prédiction.[3]

Comme déjà remarqué dans §5.3.3.3, on voit que la variation dans la plupart des classes est grande, ce qui rend l'utilisation de la carte pour la prédiction difficile. Pour résoudre ce problème, l'étape suivante est nécessaire.

---

2. Pour la carte rectangulaire, on n'a pas inclus le centroïde **0**.
3. Ceci, en combinaison avec le fait que la visualisation est plus difficile, est la raison pour laquelle on n'a pas essayé les cartes hexagonales (où chaque centroïde a six plus proches voisins).

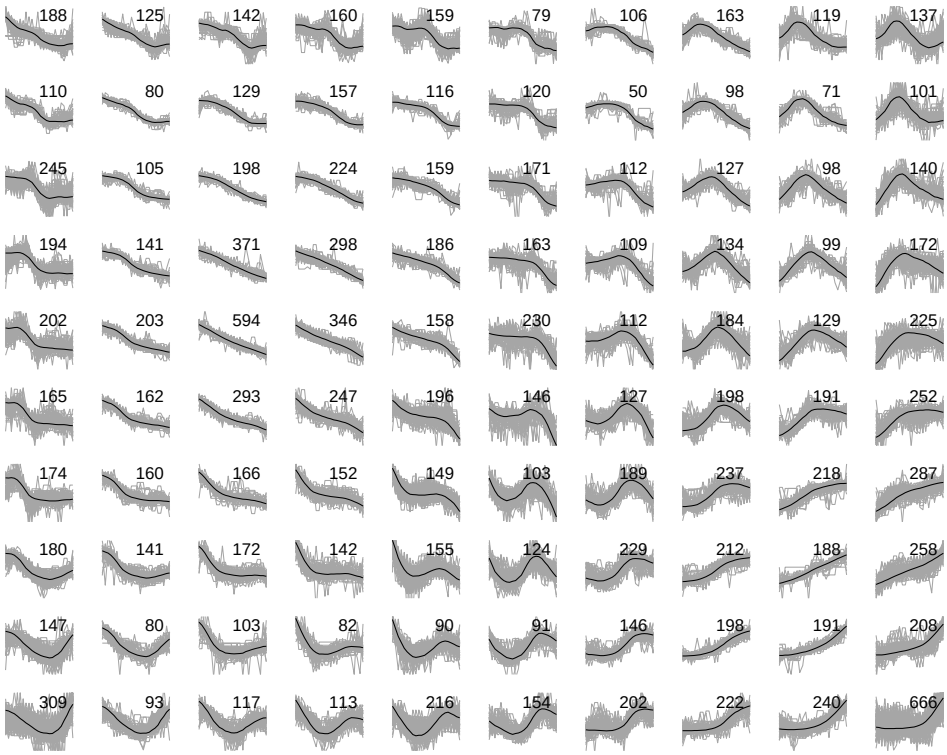

FIGURE 6.1: Une carte rectangulaire.

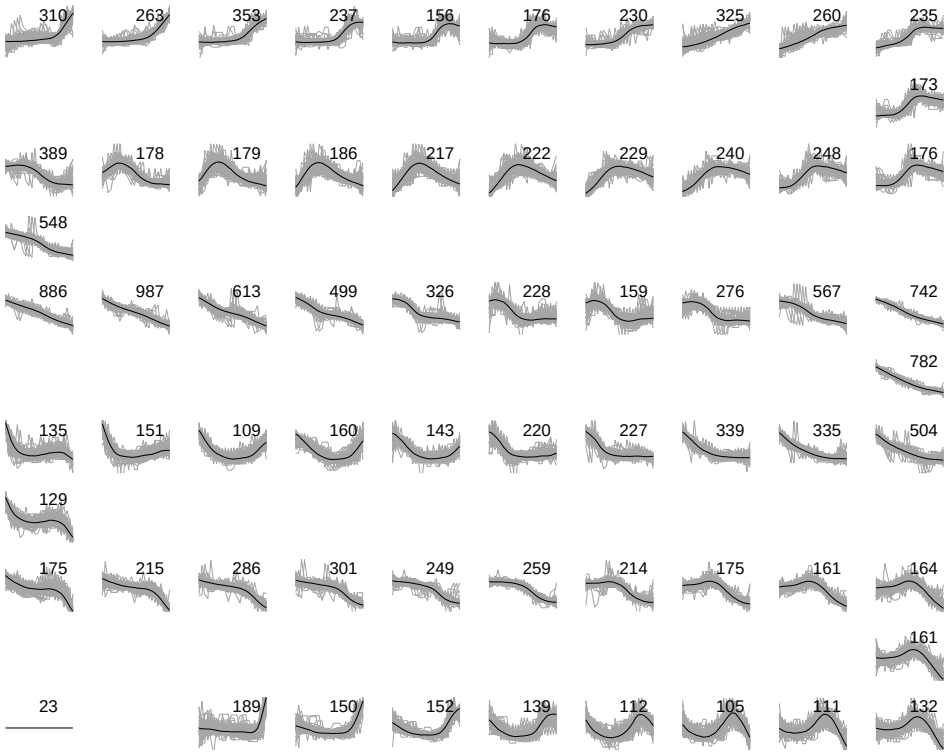

FIGURE 6.2: Une ficelle.

6.2.2.2 Analyse

L'étape de l'analyse a pour but de définir des situations pour lesquelles la variation dans chacune des classes est assez petite. Il est clair qu'il y a plusieurs méthodes d'analyse qu'on pourrait concevoir.

Nous proposons ici une analyse qui est basée sur les moyennes μ_k et les écart types σ_k des courbes de débit $\boldsymbol{Q}_k$. On construit un graphique σ_k/μ_k où on représente chaque $\boldsymbol{Q}_k$ dans l'ensemble d'apprentissage comme un point (μ_k, σ_k). Nous avons fait une distinction entre les points pour lesquels

- le maximum de $\boldsymbol{Q}_k$ dépasse le seuil d'alerte c_A,
- le maximum de $\boldsymbol{Q}_k$ dépasse le seuil de pré-alerte c_{PA} (mais pas c_A),
- aucun de ces deux seuils est dépassé.

Ce graphique est montré dans la figure 6.3. On y a utilisé la notation

$$\boldsymbol{Q}_k = (Q_{k,1}, \ldots, Q_{k,j}, \ldots, Q_{k,n}).$$

Comme on pourrait s'y attendre, les trois types de courbes de débit se trouvent surtout dans trois régions différentes du graphique.

Sur base de ce graphique on peut proposer des «situations du terrain». Un exemple d'une telle «situation du terrain» est la suivante :[4]

> Dans les vingt-quatre heures suivantes, le débit croît tellement qu'il dépassera le seuil d'alerte.

Plus formellement dit, ceci veut dire qu'on estime que le point $(\hat{\mu}_0, \hat{\sigma}_0)$ tombe dans la région $\max_j Q_{k,j} \geq c_A$ et le point $(\mu_{-24}, \sigma_{-24})$ de la courbe précédente se trouve dans la région $\max_j Q_{k,j} < c_A$.

On ne peut pas utiliser un critère comme $\max_j \hat{Q}_{0,j} \geq c_A$ pour la prédiction, vu qu'on ne connaît pas $\hat{\boldsymbol{Q}}_0$. Le critère pratique qu'on utilisera pour délimiter la région $\max_j Q_{k,j} \geq c_A$ sera de la forme

$$\hat{\mu}_0 + x\hat{\sigma}_0 = c_A, \qquad x \in \mathbb{R}^+.$$

Pour garantir que tous les points de l'ensemble d'apprentissage pour lesquels $\max_j Q_{k,j} \geq c_A$ sont inclus on prend $x = 2,34$. On a ajouté la droite correspondante

$$\mu_k + 2,34\sigma_k = c_A$$

dans la figure 6.3, de même que les droites $\mu_k = c_A$ et $\mu_k = c_{PA}$.

On voit ici, que même avant d'avoir une prédiction du profil, on peut déjà obtenir un résultat utile quand on a prédit $(\hat{\mu}_0, \hat{\sigma})$. A ce moment on connaît en effet la situation à considérer.

Dans la suite de ce chapitre, nous prenons comme situation celle déduite de la «situation du terrain» de dépassement du seuil d'alerte.[5] C'est à dire, la

4. Dans §A.2 nous proposons un ensemble complet de situations.
5. On a aussi examiné le dépassement du seuil de pré-alerte et on a obtenu des résultats semblables.

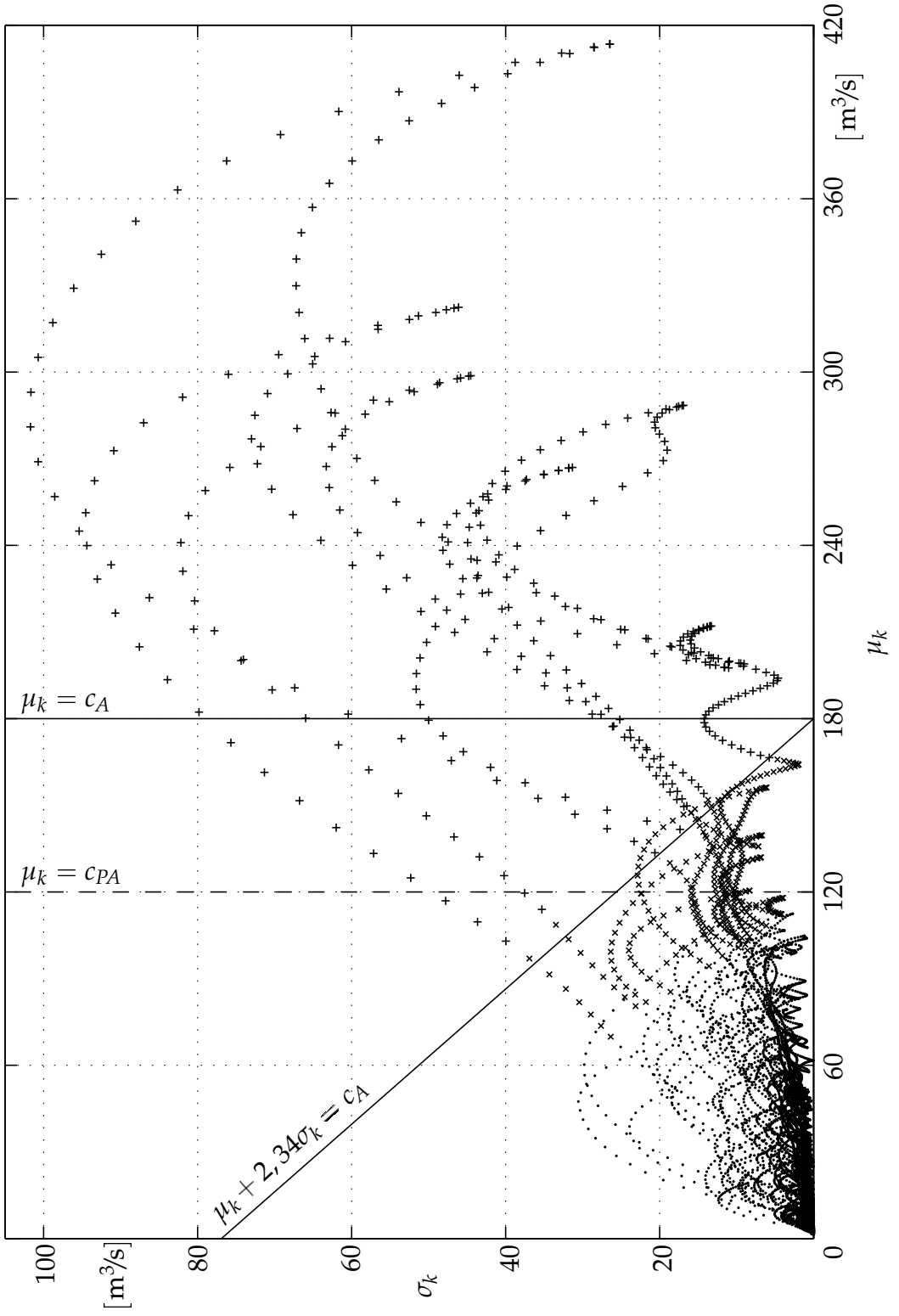

FIGURE 6.3: Graphique σ_k / μ_k pour les courbes de débit $\boldsymbol{Q}_k$, où on a utilisé
- un $\cdot$ quand $\max_j Q_{k,j} < c_{PA}$,
- un $\times$ quand $c_{PA} \leq \max_j Q_{k,j} < c_A$,
- un $+$ quand $c_A \leq \max_j Q_{k,j}$.

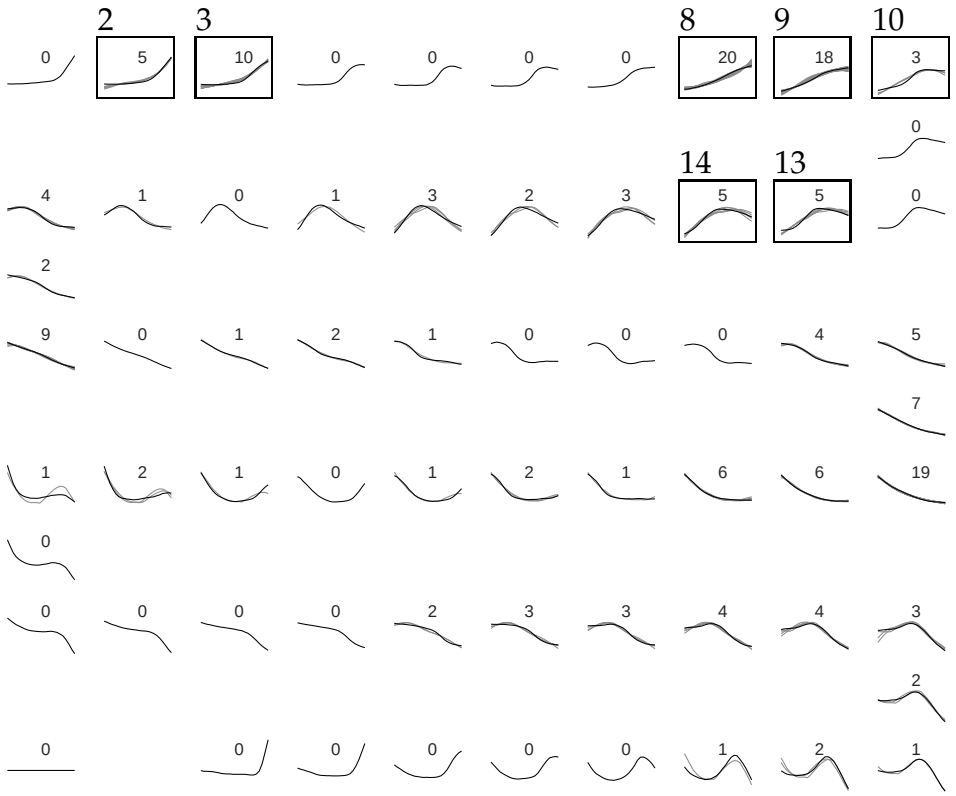

FIGURE 6.4: Variations réduites pour la situation considérée.

situation pour laquelle $\hat{\mu}_0 + 2,34\hat{\sigma}_0 \geq c_A$ et $\max_j Q_{-24,j} < c_A$. Pour montrer que la variation dans les classes pour cette situation est assez petite, on donne (dans la figure 6.4) la carte auto-organisatrice utilisée,[6] avec dans les classes les profils $\boldsymbol{\pi}_k$ de l'ensemble d'apprentissage pour lesquels $\mu_k + 2,34\sigma_k \geq c_A$. Les profils correspondants à la situation à laquelle on s'intéresse se trouvent dans les classes encadrées (avec le numéro de la classe). On voit qu'à la réduction des variations, il s'ajoute le fait que le nombre de classes à considérer est fort réduit (voir classes encadrées sur la figure). Les centroïdes correspondants à ces classes sont, pas étonnamment, tous croissants.

6.2.2.3 Construction de la procédure

Pour illustrer l'étape de la construction d'une procédure de prédiction de profil, on prend donc l'exemple d'une procédure pour la situation de dépassement du seuil d'alerte introduit ci-dessus dans §6.2.2.2.

La première chose qu'on fait est d'examiner les différentes transitions du profil $\boldsymbol{\pi}_{k-24}$ au profil $\boldsymbol{\pi}_k$, c'est à dire les transitions entre les classes auxquelles ils appartiennent. Dans la figure 6.5 on montre ces transitions et leurs fré-

6. Cette carte a 63 centroïdes et est construite sur base de 10 époques de 17500 entrées $\boldsymbol{\pi}_k$. On a aussi essayé une carte avec 41 centroïdes, ce qui donnait des résultats semblables.

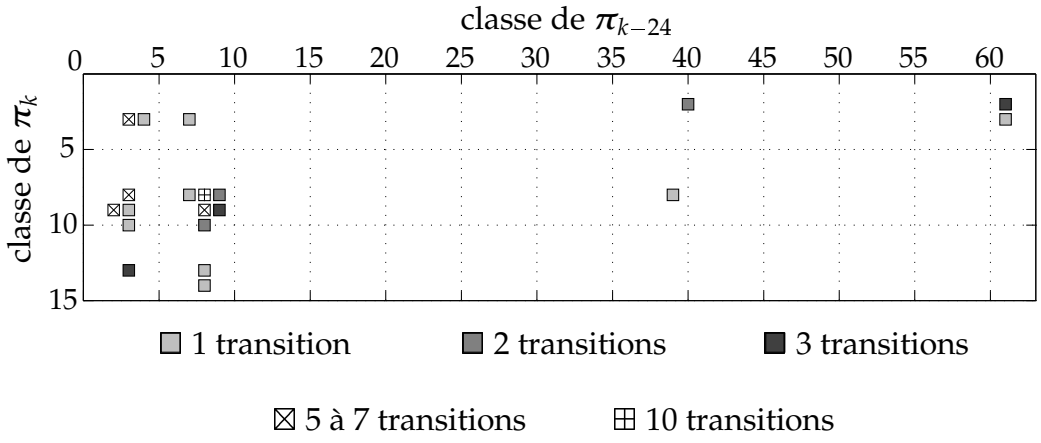

FIGURE 6.5: Transitions entre les classes pour des profils successifs.

quences absolues dans l'ensemble d'apprentissage. Contrairement à ce qu'on a rencontré pour la prédiction de la consommation d'électricité cette information[7] n'est pas suffisante pour prédire le profil, car ces profils peuvent être trop différents (il existe par exemple à partir de la classe 3 des transitions non négligeables vers presque toutes les classes possibles pour la situation étudiée : 3, 8, 9, 10 et 13, voir la figure 6.5).

On doit alors utiliser d'autre données. Nous allons examiner la relation des transitions (vers la *classe de destination* i de $\boldsymbol{\pi}_k$) avec une grandeur G quelconque. Les grandeurs que nous avons choisies sont la dernière mesure de débit Q_0 et une somme de pluies passées et de prévisions météorologiques $\sum_{k=-5}^{13} PB_k$.[8] Dans la figure 6.6 on a visualisé ces deux relations avec des graphiques. Chaque point (i, G) dans les graphiques correspond à un profil $\boldsymbol{\pi}_k$ de l'ensemble d'apprentissage. Pour chaque *classe d'origine* (la classe de $\boldsymbol{\pi}_{k-24}$), on a noté le numéro de la classe à la valeur moyenne de la grandeur (pour les profils dans la classe i).

C'est sur base des graphiques dans les figures 6.5 et 6.6 qu'on construit la procédure de prédiction. Cette procédure se déroule comme suit :

1. avec le graphique dans la figure 6.5 et la classe d'origine on identifie les transitions possibles,
2. à l'aide des graphiques de la figure 6.6 et les valeurs des grandeurs on élimine les transitions pour lesquelles il n'existe pas de précurseurs dans l'ensemble d'apprentissage,[9]

7. Cette information est équivalente aux probabilités de transition vue dans §5.3.3.2.
8. Q_0 est choisi comme indication du régime initiale et $\sum_{k=-5}^{13} PB_k$ est choisie pour déterminer le type d'évolution dans ce régime.
9. Nous avons trouvé que la somme de pluies était la grandeur la plus importante pour la prédiction.

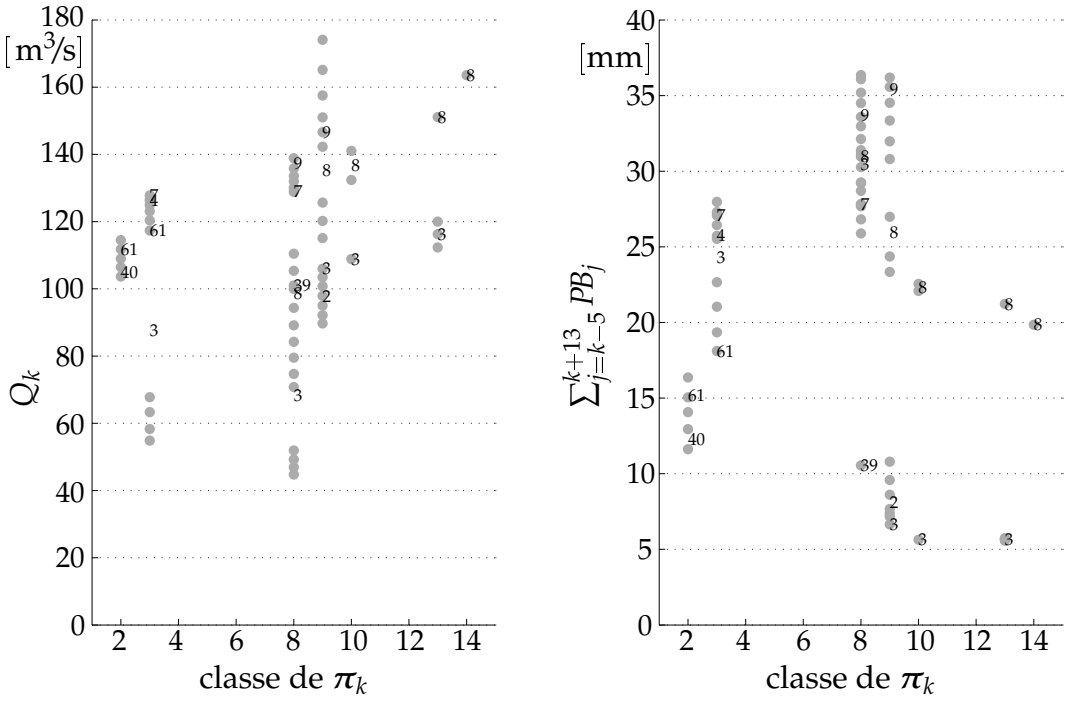

FIGURE 6.6: Relation des transitions avec des grandeurs.

3. s'il reste encore de l'ambiguïté, c'est à dire, plusieurs transitions possibles, on prend celle qui à la plus grande fréquence dans l'ensemble d'apprentissage.

L'ambiguïté dont on parlait se produit surtout entre transitions vers classes voisines. Grâce au fait qu'on a utilisé une carte auto-organisatrice pour construire les classes (les centroïdes des classes voisins sont similaires), l'erreur commise reste limitée.

Nous donnons un exemple[10] illustrant la procédure qu'on vient de décrire. Nous voulons prédire le profil π_0 pour lequel la classe du profil précédent π_{-24} est 3, $Q_0 = 102{,}5\,\mathrm{m}^3/\mathrm{s}$ et $\sum_{k=-5}^{13} PB_k = 34{,}7\mathrm{mm}$. Les transitions possibles sont vers les classes $3, 8, 9, 10$ et 13, mais il n'y a pas de transitions vers 2 et 14. Sur base du dernier débit toutes ces classes restent encore plausibles. Ceci est déterminé comme suit :

– Regardez pour chaque classe i de $\hat{\pi}_0$ encore possible si le point (i, Q_0) se trouve assez proche d'un point de l'ensemble d'apprentissage. Pour $i = 13$ on considère par exemple que c'est assez proche. La classe $i = 3$ est encore retenue car (i, Q_0) tombe entre deux groupes de points de l'ensemble d'apprentissage.

10. Les exemples qu'on donne dans cette section ont été choisis dans l'ensemble de validation et donnaient des bonnes prédictions.

– Comparez la position de (i, Q_0) avec la position de la moyenne pour la classe d'origine (ici 3). Si l'écart entre les deux est assez petit, la transition reste possible.

Sur base de la somme des pluies, uniquement la classe 8 reste plausible. C'est à dire, pour les classes $i = 8$ et $i = 9$ il y a des points qui sont assez proche de $(i, \sum_{k=-5}^{13} PB_k)$, mais l'écart avec la moyenne reste assez petit uniquement pour $i = 8$.

Jusqu'ici on a supposé qu'on pouvait trouver des précurseurs dans l'ensemble d'apprentissage. Il est possible de faire des *généralisations*, ce que nous illustrons avec un deuxième exemple. Nous voulons prédire le profil $\boldsymbol{\pi}_0$ pour lequel la classe du profil précédent $\boldsymbol{\pi}_{-24}$ est 1, $Q_0 = 78{,}5\,\mathrm{m}^3/\mathrm{s}$ et $\sum_{k=-5}^{13} PB_k = 33{,}5\mathrm{mm}$. Comme (le centroïde de) la classe 1 est similaire aux (centroïdes des) classes 2 et 3, on examinera les transitions à partir de ces classes. On trouve de nouveau la classe 8 comme classe de destination. Cet exemple montre qu'on doit être capable de faire du regroupement pour pouvoir généraliser. Ceci est possible grâce à la propriété «topologique» des cartes auto-organisatrices : les centroïdes qui sont proches l'un de l'autre dans la structure sont similaires.

6.2.2.4 Exemples de profils prédits

Jusqu'à maintenant, on a vu comment de prédire la classe i du profil $\boldsymbol{\pi}_0$. La prédiction du profil est maintenant réduit à choisir un profil qui représente cette classe. Le représentant trivial est le centroïde $\boldsymbol{c}_i$ de la classe. Comme on l'a vue à la fin de §5.3.3.3, la deuxième option est de prendre la moyenne des profils de l'ensemble d'apprentissage se trouvant dans la classe i pour la situation étudiée.

Dans la figure 6.7 on a montré deux profils à prédire (ligne continue) et les prédictions correspondantes : le centroïde (ligne interrompue) et la moyenne de la classe (ligne pointillée). L'abscisse (ici est dans les figures 6.8 et 6.9) correspond aux composantes du profil. Pour presque toutes les prédictions qu'on a faites, la moyenne de la classe était meilleure que le centroïde. Nous choisissons donc comme prédiction $\hat{\boldsymbol{\pi}}_0$ la moyenne de la classe i pour la situation étudiée.

Des profils prédits, il y avait 10% qui n'était pas du tout similaire aux vrais profils.

6.2.3 Prédictions totales

On a déjà vu à la fin de §6.2.1 qu'on a pas à disposition les prédictions $\hat{\mu}_0$ et $\hat{\sigma}_0$ pour les courbes de débit. Pour montrer des prédiction totales on a alors utilisé les vraies valeurs μ_0 et σ_0. C'est à dire,

$$\check{\boldsymbol{Q}}_0 = \sigma_0 \hat{\boldsymbol{\pi}}_0 + \mu_0 \mathbf{1}.$$

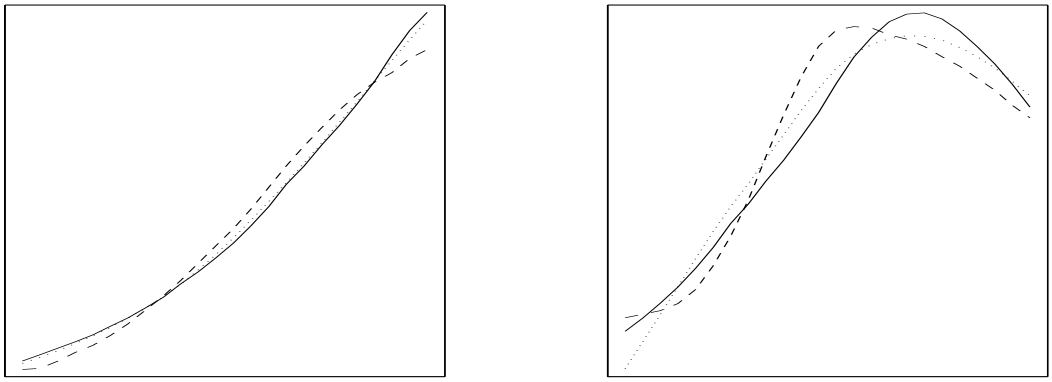

FIGURE 6.7: Prédiction de deux profils de l'ensemble de validation.

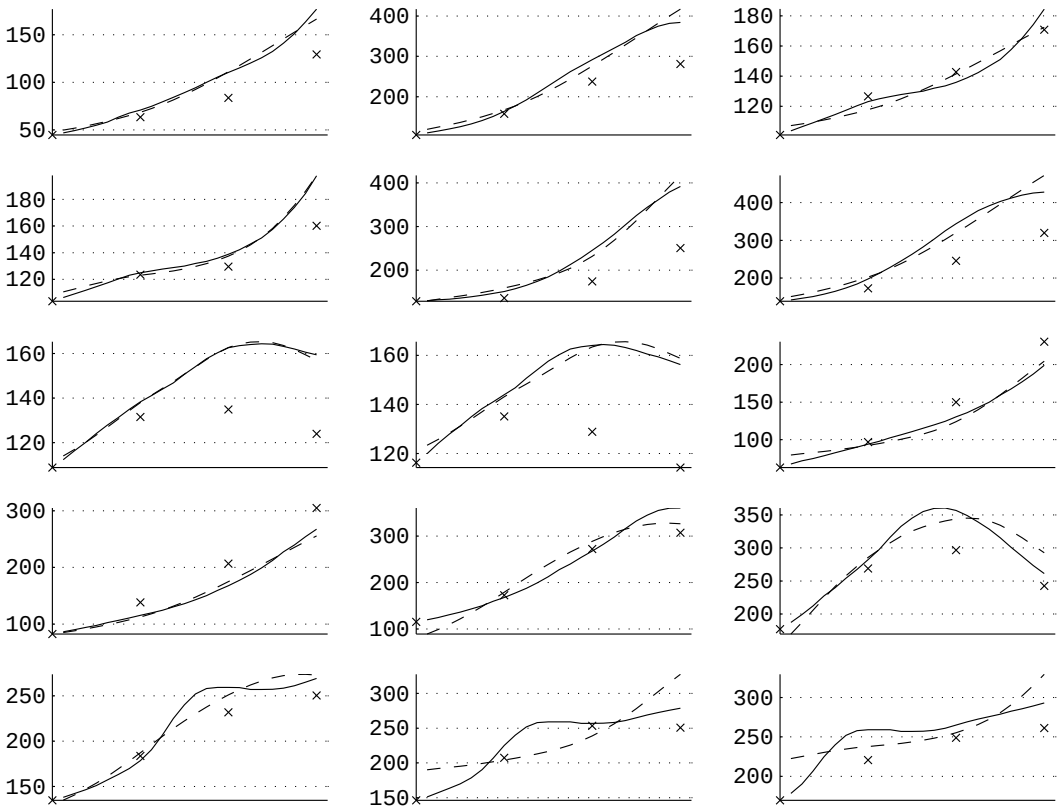

FIGURE 6.8: Exemples de prédictions totales.

Une quinzaine de ces prédictions de $\check{\boldsymbol{Q}}_0$ est montrée dans la figure 6.8. En ordonnée, on a mis le débit en m^3/s. Les lignes continues sont les vraies courbes $\boldsymbol{Q}_0$, les lignes interrompues sont les prédictions (pour les $\times$, voir §6.3). Les huit premiers exemples viennent de l'ensemble d'apprentissage et les sept derniers de l'ensemble de validation.

On voit que la plupart de ces prédictions, qui doivent être vu comme un test de la prédiction des profils, est satisfaisante. Les deux dernières prédictions sont clairement de mauvaise qualité, mais on peut les améliorer en modifiant la procédure de prédiction (voir §6.4).

6.3 Comparaison avec Hydromax

Avant de pouvoir faire une comparaison des résultats, il est utile de donner un peu d'information pratique sur Hydromax. Pour le bassin de la Lesse à Gendron h vaut 8h et le régresseur, qui est différent pour chaque bassin, consiste de trois mesures de débits Q_k et trois valeurs de la pluie brute PB_k. Si on veut prédire Q_h, les indices de ces éléments du régresseur sont $k = 0, -8, -16$. Les prédictions successives (à court terme, puis à long terme) de Hydromax sont séparées de 8h et on utilise des splines comme méthode d'interpolation pour les points intermédiaires voulus.[11]

Dans la figure 6.8 nous avons aussi ajouté des prédictions produites par Hydromax. Avec un « $\times$ », on a indiqué respectivement la dernière mesure Q_0, la prédiction à court terme Q_8 et deux prédictions à long terme Q_{16} et Q_{24}. On voit que la plupart de nos «prédictions» $\check{\boldsymbol{Q}}_0$ sont meilleures que les prédictions de Hydromax. Ceci doit être interprété avec caution. Si les erreurs pour la prédiction de la moyenne et l'écart type des courbes de débit sont suffisamment petites, on peut conclure, pour la situation étudiée, que notre méthode donne des prédictions comparables ou meilleures que celles produites par Hydromax.

Pour Hydromax, l'erreur relative $|1 - \frac{\hat{Q}_k}{Q_k}|$ est de l'ordre de 10% pour $k = 8$h et de l'ordre de 20% pour $k = 24$h. Cette augmentation de l'erreur avec le temps est visible pour les prédictions dans la figure 6.8. Pour notre méthode, l'erreur commise dépend de la variation dans la classe correspondant au profil prédit (ou la variation dans une groupe de classes, si on a fait de la généralisation). Cette erreur sera donc plus ou moins constante pour tout les composants de la courbe. Ceci est illustré dans la figure 6.9, où la vraie courbe (ligne continue) est comparé aux prédictions correspondant au «maximum» et au «minimum» de la variation (lignes pointillées).

11. La même méthode d'interpolation peut bien sûr aussi être appliquée aux résultats obtenus avec notre méthode.

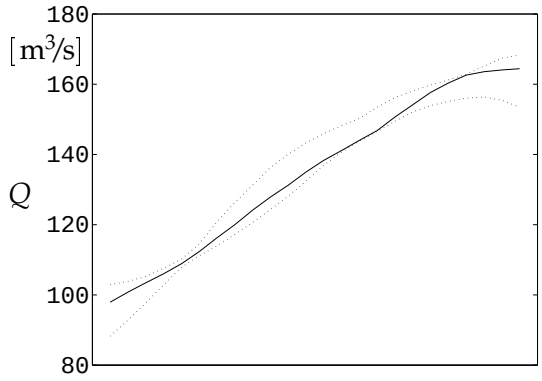

FIGURE 6.9: L'erreur due à la variation dans les classes.

Si on compare la complexité des deux modèles, il faut conclure que Hydromax est plus simple que notre méthode.[12] Hydromax utilise deux sous-modèles, avec trois paramètres pour la fonction de production et une dizaine de paramètres pour le modèle ARX. Notre méthode utilise trois sous-modèles, chacun avec des centaines de paramètres (rappelons que chacun des centroïdes pour la carte auto-organisatrice contient déjà 24 paramètres).

6.4 Suggestion d'améliorations

C'est la première fois que notre méthode de prédiction est appliquée, il est donc clair, vue les limitations de temps, que pas toutes les idées initiales ont été mises en œuvre. On n'a aussi pu tester qu'une minorité des choix possibles de §5.3.5. En plus, ce travail même nous a aidé à formuler de nouvelles idées.

Parlons d'abord des choix possibles. Il est clair qu'on aura pu choisir beaucoup de différentes longueurs des courbes de débit n et intervalles de temps h. Dans les figures 6.1, 6.2 et 6.4 on voit que la plupart des centroïdes des classes ont une allure linéaire sur l'ordre de quelques (3 à 4) heures. En gardant nh constant et en augmentant h, on pourrait donc fort réduire n. Ceci pourrait permettre de diminuer la taille des cartes auto-organisatrices utilisées. Ceci aurait comme conséquence que la densité de profils dans $\mathbb{R}^n$ serait fort augmentée, ce qui est positif pour la formation des cartes.

Pour les prédictions des profils faites dans le cadre de cette mémoire, la procédure n'était pas automatisée. Si on espère que notre méthode soit utile pour la pratique, ceci est nécessaire. Une option est de construire un arbre de décision. Chaque nœud correspond à certaines classes de destination. Le nœud initial correspond à toutes les classes et chacun des nœuds terminaux à une classe spécifique. Les branches correspondent aux décisions faites sur base des grandeurs. Ceci est illustré dans la figure 6.10. Les difficultés sont le choix de

12. Nous supposons déjà connue la fonction de production utilisé dans Hydromax (§5.2.1).

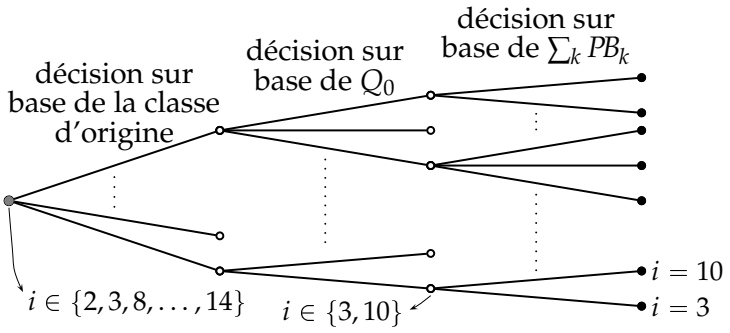

FIGURE 6.10: Illustration de l'automatisation avec un arbre de décision.

l'ordre des grandeurs dans l'arbre et la formulation des règles de décision. Une deuxième option pour l'automatisation est d'utiliser une méthode de classification supervisée.[13] Pour cette classification supervisée on associe une classe à chaque classe de destination d'une transition. La classification se fait sur base des grandeurs considérées et les classes à l'origine des transitions. Une procédure de prédiction du profil automatisée nous permettra aussi d'utiliser des cartes auto-organisatrices avec une structure rectangulaire ou hexagonale.

Dans les figures 6.5 et 6.6 on voit que le nombre de données sur lequel on s'est basé pour faire des prédictions est relativement limité. Dans l'ensemble de validation on a rencontré des cas pour lesquels il était difficile, même avec généralisation, de trouver une prédiction. Pour éviter ce problème il serait très utile d'avoir à disposition un ensemble d'apprentissage plus grand.

Pour plusieurs prédictions qu'on a fait, la prédiction du premier débit $\hat{Q}_1$ était clairement trop différent de Q_0 pour être plausible. On pourrait dire que la continuité entre les courbes de prédiction n'était pas respectée. On pourrait tenir compte de cette information en définissant la procédure de prédiction de profil de telle façon qu'elle donne plusieurs candidats. Le candidat qui respecte le plus la continuité est alors choisi. On a appliqué cette méthode au 10% mauvaises prédictions dont on parlait dans §6.2.2.4, et les nouvelles prédictions étaient meilleures. Pour faciliter ce choix entre candidats, on pourrait faire des prédictions $\hat{\boldsymbol{Q}}_0 = (\hat{Q}_0, \ldots, \hat{Q}_{[n-1]h})$ et pas $\hat{\boldsymbol{Q}}_0 = (\hat{Q}_h, \ldots, \hat{Q}_{nh})$. On pourrait dans ce cas comparer $\hat{Q}_0$ avec la vraie valeur Q_0.

6.5 Bibliographie

Les calculs pour les réseaux à RBF dans le cadre de la section 6.2.1 ont été faits par A. Lendasse avec MATLAB® et le logiciel «RBFm» (voir [22]).

Les prédictions faites par Hydromax utilisées dans la figure 6.8 ont été fournies par L. Moens. Pour la section 6.3 on a consulté [1].

13. Des exemples de ce type de méthodes sont par exemple donnés dans [3, *Ch.6*].

Chapitre 7

Conclusions

7.1 Méthode

Le but de ce mémoire était d'étudier une méthode pour prédire le débit de rivières. Cette étude consistait à généraliser une méthode déjà utilisée pour la prédiction de la consommation d'électricité journalière. Principalement à cause de la différence entre l'évolution de la grandeur à prédire (figure 5.3), cette généralisation n'était pas triviale. Les modifications les plus importantes concernaient la prédiction des profils. Elles étaient l'introduction des situations et l'utilisation des variables exogènes dans la procédure de prédiction (voir §5.3.3.3).

La méthode que nous avons proposée est prometteuse, mais n'est clairement pas encore mise au point. Pour ça, il y a encore trop de choix à effectuer (voir §5.3.5) et la détermination des situations et des procédures de prédiction du profil n'est pas encore assez automatisée.

Si nous comparons notre méthode avec Hydromax, la différence la plus fondamentale est que nous n'utilisons pas de fonction de production. Ce manque d'utilisation de connaissances théoriques du domaine d'application implique que notre méthode pourrait être facilement adaptée à d'autres domaines. C'est à dire, cette méthode ne doit pas seulement être vue comme une méthode de prédiction de débit de rivières, mais comme une méthode de prédiction qui peut avoir plusieurs applications.

7.2 Résultats

Ce qui manque le plus dans ce mémoire, dû à la limitation du temps, sont des prédictions complètes. C'est à dire, des prédictions constituées de moyennes prédites, d'écart types prédits et de profils prédits. Cependant, les résultats obtenus pour chacune de ces trois parties des courbes de débit étaient satisfaisants.

Dans §6.2.1, nous avons montré qu'avec des modèles de régression non linéaires, basé sur des réseaux à fonctions radiales de base, on peut obtenir des prédictions pour la moyenne et l'écart type avec une erreur limitée.

Dans §6.2.2, il est devenu clair qu'on peut construire des procédures de prédiction du profil qui donnent des résultats satisfaisants.

Il est intéressant de noter qu'un résultat intermédiaire de ces procédures, la situation, peut déjà être utile en soi. Il est par exemple très utile de savoir qu'il est très probable que le débit va dépasser le seuil d'alerte.

Il faut rester prudent et ne pas automatiquement généraliser les résultats trouvés dans ce mémoire pour la prédiction des profils (pour la situation de dépassement du seuil d'alerte,[1] §6.2.2.3) vers la prédiction des profils pour toutes les situations. Cependant, nous pouvons faire quelques remarques pour les situations concernant les courbes de débit avec des petites moyennes et des petits écart types. Pour ces situations, le nombre de données disponibles est plus grand (voir la figure 6.3), ce qui facilite la prédiction. L'augmentation correspondante de la variation dans les classes a moins d'importance car l'écart type est petit.

De la comparaison entre nos résultats et celles de Hydromax, nous concluons que notre méthode peut devenir une méthode fiable pour la prédiction de débit de rivières, si on continue sa développement. Le fait que notre méthode est plus complexe (on utilise beaucoup plus de paramètres), doit seulement être vu comme un désavantage quand les calculs prennent trop de temps avec les outils de calcul disponibles.

7.3 Suggestions pour la recherche

L'expérience acquise pendant le travail pour ce mémoire nous permet d'évaluer l'approche prise et de formuler des suggestions pour une continuation de la recherche. Cette approche était de fixer une application (la prédiction de débit de rivières) et de fixer une méthode de prédiction à généraliser (celle de [2]).

Du point de vue de la prédiction de débit, on a essayé de développer une méthode du type boîte-noire pour un système hydrologique spécifique, un bassin versant. Si ceci est le but, il est mieux de d'abord considérer plusieurs candidats de méthodes et ensuite d'examiner celles qui sont les plus aptes. Pour cette direction de recherche, les chapitres 2 et 4 et la section 5.2 peuvent être utiles.

Du point de vue de la généralisation de la méthode, on a essayé d'appliquer une méthode développée pour une grandeur avec une sorte de périodi-

1. On ne l'a pas montré dans ce mémoire, mais on a aussi obtenu des résultats semblables pour la situation de dépassement du seuil de pré-alerte.

cité journalière à une grandeur sans une telle périodicité. Si ceci est le but, il est mieux de d'abord considérer des grandeurs d'un type entre les deux décrit ci-dessus pour mieux comprendre quelles modifications sont nécessaires.[2] Pour cette direction de recherche les sections 3.3 et 5.3 sont très utile. Un des buts de cette recherche devrait être de trouver les caractéristiques des grandeurs auxquelles la méthode peut être appliquée.

2. Pour bien choisir ces grandeurs, il faut bien sûr d'abord identifier les différences entre les deux types décrits ci-dessus.

Annexe A

Remarques complémentaires

A.1 Construction d'une SOM sur base des profils

Dans cette section, nous regardons de plus près pourquoi nous avons construit les cartes auto-organisatrices (pour la classification non supervisée des profils $\boldsymbol{\pi}_k$) avec la méthode de §3.3.5. Nous nous posons aussi la question s'il y a besoin d'un prétraitement des données.

On sait que les profils $\boldsymbol{\pi}_k$ sont normalisés, c'est à dire,

$$\|\boldsymbol{\pi}_k\| = \sqrt{\sum_{j=1}^{n} \pi_{k,j}^2} = \frac{1}{\sigma_k} \sqrt{\sum_{j=1}^{n} [Q_{k+jh} - \mu_k]^2} = n,$$

donc toutes les normes[1] des profils sont égales. Il est alors naturel d'utiliser les cartes auto-organisatrices à produit scalaire définies dans §3.3.5 avec (3.4), comme la norme des centroïdes ne resterait pas constante si on utiliserait l'équation (3.3).

A cette normalisation s'ajoute le fait que les profils sont centrés, c'est à dire,

$$\sum_{j=1}^{n} \pi_{k,j} \propto \sum_{j=1}^{n} [Q_{k+jh} - \mu_k] = 0.$$

La conséquence est que les profils et les centroïdes se trouvent sur un sous-ensemble de l'hypersphère S^n. On l'appellera l'hypercercle C^n,[2]

$$C^n = \{\boldsymbol{u} \in S^n \subset \mathbb{R}^n \mid \sum_{j=1}^{n} u_j = 0\}.$$

On pourrait se demander si un prétraitement des courbes de débit $\boldsymbol{Q}_k$ est nécessaire. Dans §3.3.6 on a vu qu'une «normalisation $(0, 1)$» des échelles des

1. Nous utiliserons une norme euclidienne.
2. Par exemple, S^3 est un sphère et C^3 est un cercle.

propriétés est utile si on veut que toutes les composantes de $\boldsymbol{Q}_k$ aient la même importance pendant la construction de la carte. Ceci est le cas, parce qu'il n'y a pas de composante fixée de $\boldsymbol{Q}_k$ qui est plus importante qu'une autre. La façon de construire les $\boldsymbol{Q}_k$ (voir figure 5.3) garantit que les échelles des propriétés ont une même moyenne et un même écart type.[3] Ceci peut être compris quand on se réalise que les courbes formées d'une même composante $Q_{k,j}$ de $\boldsymbol{Q}_k$ (pour tout k) sont identiques à une translation près. Il s'en suit qu'un prétraitement des courbes de débit n'est pas nécessaire.

A.2 Situations pour la prédiction du profil

Dans cette section, nous examinons d'une façon plus générale que dans §6.2.2.2 comment les situations sont définies. Le point de départ est de nouveau la figure 6.3.

Sur base de ce graphique on propose les «situations du terrain» suivantes.

(i) On estime que le point $(\hat{\mu}_0, \hat{\sigma}_0)$ tombe dans la région $\max_j Q_{k,j} \geq c_A$ et
  - la dernière courbe de débit $\boldsymbol{Q}_{-24}$ se trouve dans la même région,
  - $\boldsymbol{Q}_{-24}$ se trouve dans une région pour laquelle $\max_j Q_{k,j} < c_A$.

(ii) On estime que $(\hat{\mu}_0, \hat{\sigma}_0)$ tombe dans la région $c_{PA} \leq \max_j Q_{k,j} < c_A$ et
  - $\boldsymbol{Q}_{-24}$ se trouve dans une région pour laquelle $\max_j Q_{k,j} \geq c_A$,
  - $\boldsymbol{Q}_{-24}$ se trouve dans la même région,
  - $\boldsymbol{Q}_{-24}$ se trouve dans une région pour laquelle $\max_j Q_{k,j} < c_{PA}$.

(iii) On estime que $(\hat{\mu}_0, \hat{\sigma}_0)$ tombe dans la région $\max_j Q_{k,j} < c_{PA}$ et
  - $\boldsymbol{Q}_{-24}$ se trouve dans une région pour laquelle $\max_j Q_{k,j} \geq c_{PA}$,
  - $\boldsymbol{Q}_{-24}$ se trouve dans la même région.

On crée alors une situation pour chaque transition dans une région ou entre deux régions.

On ne peut pas utiliser un critère comme $\max_j \hat{Q}_{0,j} < c_{PA}$ pour la prédiction de $\boldsymbol{\pi}_0$, vu qu'on ne connaît pas $\hat{\boldsymbol{Q}}_0$. Les critères pratiques qu'on utilisera pour délimiter les régions seront de la forme

$$\hat{\mu}_0 + x\hat{\sigma}_0 = c, \qquad x \in \mathbb{R}^+,$$

avec c égal à c_{PA} ou c_A. On pourra ajouter quelques exemples de ces droites dans la figure 6.3 pour illustrer qu'elles peuvent être utilisées pour approximativement délimiter les régions.

---

3. Remarquons que pour l'application de la prédiction de la consommation d'électricité journalière trouvée dans la littérature, les moyennes et les écarts types des échelles ne sont pas égaux. Contrairement à notre situation, les composantes de $\boldsymbol{C}_k$ ne sont pas nécessairement aussi important pour la construction de la carte. Par exemple, pour faire la classification, les petites variations des minima peuvent être considérées moins important que les variations des maxima.

Les situations pratiques ainsi définies ne doivent pas être mutuellement exclusives. C'est à dire, si la borne inférieure de la région $\max_j Q_{k,j} \geq c_A$ est par exemple définie avec un $x = 2,34$ (ce qu'on a fait), on peut définir la borne supérieure de la région $c_{PA} \leq \max_j Q_{k,j} < c_A$ avec un $x < 2,34$. Si un point $(\hat{\mu}_0, \hat{\sigma}_0)$ tombe dans deux régions, on peut faire deux prédictions $\hat{\boldsymbol{\pi}}_0$ du profil $\boldsymbol{\pi}_0$, qui seront très similaires pour des bonnes procédures de prédiction.

Annexe B

Algorithmes et calculs

Tous les calculs ont été faits dans l'environnement du logiciel MATLAB® de MathWorks, Inc. et les programmes écrite dans cet environnement ont été testé sur les versions 5.3 et 6.1. Même que l'efficacité (temps de calcul et capacité de stockage) des programmes écrit dans cet environnement n'est pas optimal,[1] la facilité d'emploi (fonctions prédéfinies et outils de visualisation disponibles) en combinaison avec la légèreté relative[2] des simulations en font un outil apte.

Nous donnerons dans cet annexe des versions documenté des programmes utilisées pour les différents calculs faits dans le cadre du mémoire. Nous n'inclurons pas les programmes réglant la visualisation. Les instructions des calculs pour lesquels nous n'avons pas fait de programme, n'ont aussi pas été incluses.

B.1 Cartes auto-organisatrices

Les programmes montrées ici sont l'implémentation de la théorie de la section §3.3.

Les structures de données utilisées pour représenter la liste d'entrées et la structure sont des «cell arrays». D'abord nous définissons quelques constantes pour référencer les cellules d'un «cell array».

`constants.m`

```
01 nud=1;   Le nombre de données (m ∈ ℕ).
02 did=2;   La dimension des données (n ∈ ℕ).
03 lid=3;   La liste de données (matrice m × n).
04 tyi=4;   Le type de quantification vectorielle.
```

Les différentes types de quantification vectorielle programmés sont la quantification vectorielle classique (`cvq`) et les cartes auto-organisatrices à voisinage abrupt (ficelle :

1. Sous-optimal par rapport aux programmes qu'on pourrait écrire en C ou Fortran, par exemple.
2. Le calcul le plus lourd (construction d'une carte auto-organisatrice avec cent centroïdes et 17500 entrées 24-dimensionelle) fait dans le cadre de ce mémoire demandais à peu près 40 minutes de calcul sur un ordinateur moyen fabriqué en 1999.

`lin`, rectangulaire `rect` et hexagonale `hex`) et à voisinage lisse (voisinage gaussien) (`ling`, `rectg` et `hexg`).

`05 dii=5;` La dimension de l'indexation ($s \in \mathbb{N}$).

`06 lii=6;` La liste des indices des centroïdes (matrice $m \times s$).

`07 nue=4;` Le nombre d'époques.

Avant de commencer aux calculs, nous créons deux «cell arrays» qui contiennent les profils d'entrée (`x`) et les centroïdes (`m`).

`construct.m`

```
01 function [x,m]=construct(learning_data,epochs,map_type,...
                            map_dimensions)
```

Les arguments sont les données d'entrées (liste de vecteurs `learning_data`), le nombre d'époques (`epochs`), le type de quantification vectorielle (`map_type`) et le nombre de centroïdes (`map_dimensions`).

`02 constants;` Charger les constantes.

Définir le «cell array» pour les entrées :

```
03 x=cell(size([nud,did,lid]));
04   x{nud}=epochs*size(learning_data,1);
05   x{did}=size(learning_data,2);
06   x{lid}=[];
07   for j=1:epochs, x{lid}=[x{lid};learning_data]; end,
08   x{nue}=epochs;
```

Définir le «cell array» pour les centroïdes :

```
09 m=cell(size([nud,did,lid,tyi,dii,lii]));
10   switch map_type
11    case {'cvq','lin','ling','rect','rectg'},
12     m{nud}=prod(map_dimensions);
13    case {'hex','hexg'},
14     m{nud}=2*prod(map_dimensions);
15    otherwise disp('Wrong map_type given.');
16   end
17   m{did}=size(learning_data,2);
```

Les centroïdes initiales sont choisie aléatoirement dans les entrées :

```
18   m{lid}=x{lid}(ceil(x{nud}*rand(m{nud},1)),:);
19   m{tyi}=map_type;
20   switch map_type
21    case 'cvq',
22     m{dii}=0;
23     m{lii}=[];
24    case {'lin','ling'},
25     m{dii}=1;
26     m{lii}=[1:m{nud}]';
27    case {'rect','rectg'},
28     m{dii}=2;
29     [i,j]=meshgrid(1:map_dimensions(1),...
                      1:map_dimensions(2));
```

```
30     m{lii}=[reshape(i,m{nud},1),reshape(j,m{nud},1)];
31    case {'hex','hexg'},
32     m{dii}=2;
33     [i,j]=meshgrid(1:2:2*map_dimensions(1)-1,...
                      1:2:2*map_dimensions(2)-1);
34     [k,l]=meshgrid(2:2:2*map_dimensions(1),...
                      2:2:2*map_dimensions(2));
35     n=m{nud}/2;
36     m{lii}=[reshape(i,n,1),reshape(j,n,1);...
              reshape(k,n,1),reshape(l,n,1)];
37    otherwise disp('Wrong map_type given.');
38   end
```

Maintenant, nous sommes prêts pour calculer la position finale des centroïdes sur base des entrées fournis, recyclées nue fois.

learn.m

```
01 function m=learn(x,m)
02 constants;  Charger les constantes.
03 for k=1:x{nud},  Une itération par entrée.
04   xk_array=ones(m{nud},1)*x{lid}(k,:);
05   w=winner(m{lid},xk_array);  Déterminer le centroïde le plus proche.
06   l=x{nud}/x{nue};
```

Déterminer le voisinage :

```
07   N=neighbourhood(m,w,k,l)*ones(1,m{did});
```

Recalculer les positions des centroïdes avec (3.3) :

```
08   m{lid}=m{lid}+alpha(k,l)*N.*(xk_array-m{lid});
09 end
```

L'évolution de la largeur du pas est définie ci-dessous. Pour obtenir des bonnes cartes il se peut que cette évolution doit être modifiée. Une évolution linéaire par parties comme donnée ici n'est pas nécessaire.

```
10 function out=alpha(k,l)
```

Initialement, la largeur diminue vite :

```
11 if k<1000,
12   out=.9*(1/(1+k/1000));
```

Intermédiairement, la diminution ralentit :

```
13 elseif k<5*l,
14   out=.43*(1-(k-1000)/(5*l-1000))+.02;
```

Dans la phase finale, on garde la largeur constante :

```
15 else,
16   out=.02;
17 end
```

Le centroïde le plus proche de l'entrée actuelle est déterminé en cherchant la distance minimale entrée-centroïde. Il y a différentes normes possibles.

```
18 function rownumber=winner(m_array,xk_array)
19 d=euclidian(m_array,xk_array);
19 d=supnorm(m_array,xk_array);
```

```
19 d=sumnorm(m_array,xk_array);
19 d=specialnorm(m_array,xk_array);
20 [minimal_distance,rownumber]=min(d);
```

Les normes définies ci-dessus doivent bien-sûr aussi être programmées.

euclidian.m
```
01 function d=euclidian(a,b)
02 d=sum((a-b).^2,2).^(1/2);
```

supnorm.m
```
01 function d=supnorm(a,b)
02 d=max(abs(a-b),[],2);
```

sumnorm.m
```
01 function d=sumnorm(a,b)
02 d=sum(abs(a-b),2);
```

specialnorm.m
```
01 function d=specialnorm(a,b)
02 temp=abs(a-b);
03 for k=1:size(temp,2),
04  temp(:,k)=k^(2*k)*temp(:,k).^2;
05 end
06 d=sum(temp,2).^(1/2);
```

Une fonction importante est celle qui détermine le voisinage. Le voisinage associe un poids à chaque centroïde. Pour les voisinages abrupts ces poids sont 0 ou 1, pour les voisinages lisses c'est entre 0 et 1. Pour l'évolution des voisinages dans le temps, la même remarque que pour la largeur du pas est valable, des modifications pourront parfois être utiles.

neighboorhood.m
```
01 function out=neighbourhood(m,w,k,l)
02 constants;  Charger les constantes.
```

Avant de déterminer les voisinages, on centre la structure des centroïdes autour le centroïde le plus proche de l'entrée actuelle. La symétrie des structures nous permet en plus de limiter le nombre de cas à considérer.

```
03 if ~(strcmp(m{tyi},'cvq')),
04  wi_array=ones(m{nud},1)*m{lii}(w,:);
05  d=abs(m{lii}-wi_array);
06 end
```

Pour chaque type de quantification vectorielle on détermine le voisinage.

```
07 switch m{tyi}
08  case 'cvq',
```

La quantification vectorielle classique peut être vue comme un cas limite des cartes auto-organisatrices à voisinage abrupt.

```
09    temp=zeros(m{nud},1);
10    temp(w)=1;
11    out=temp;
12  case 'lin',
```

```
13    temp=d-(6-(k/l));
14    temp(w)=0;
15    out=~(temp+abs(temp));
16   case 'ling',
17    out=exp(-(m{lii}-wi_array).^2/...
              (2*(sigma(k,l,m{nud}))^2));
18   case 'rect',
19    temp=d-(6-(k/l));
20    temp(w,:)=[0,0];
21    temp=~(temp+abs(temp));
22    out=temp(:,1)&temp(:,2);
23   case 'rectg',
24    out=exp(-supnorm(m{lii},wi_array).^2/...
              (2*(sigma(k,l,sqrt(2*m{nud})))^2));
25   case 'hex',
26    temp1=sum(d,2)-(6-(k/l));
27    temp1(w)=0;
28    temp2=d(:,2)-(6-(k/l))/2;
29    temp2(w)=0;
30    out=(~(temp1+abs(temp1)))&(~(temp1+abs(temp1)));
31   case 'hexg',
32    scale=ones(length(m{lii}),1)*[1 sqrt(3)/3];
33    out=exp(-euclidian(m{lii}.*scale,wi_array.*scale).^2/...
              (2*(sigma(k,l,sqrt(2*m{nud})))^2));
34   otherwise disp('Problem with map_type given.');
35 end
```

Pour les voisinages lisses, on a utilisé une fonction gaussienne. L'évolution de l'écart type est décrit ci-dessous.

```
36 function out=sigma(k,l,n);
```

Initialement, l'écart type diminue relativement vite :

```
37 if k<1000,
38   out=.2*n*(1/(1+5*k/1000));
```

Intermédiairement, la diminution ralentit :

```
39 elseif k<5*l,
40   out=1/30*(n-15)*(1-(k-1000)/(5*l-1000))+.5;
```

Dans la phase finale, on garde l'écart type constant :

```
41 else,
42   out=.5;
43 end
```

Jusqu'à maintenant, on a regardé les programmes qui permettent de construire des cartes comme nous les ont vues dans §3.3.3. Les cartes auto-organisatrices à produit scalaire de §3.3.5 peuvent être obtenues en légèrement modifiant `learn.m`.

`dplearn.m`

```
01 function m=dplearn(x,m)
02 constants;  Charger les constantes.
```

```
03 md=m{lid};
04 for k=1:x{nud},   Une itération par entrée.
05   xk=x{lid}(k,:);
06   xk_array=ones(m{nud},1)*xk;
07   w=winner(md,xk);   Déterminer le centroïde le plus proche.
08   l=x{nud}/x{nue};
```

Déterminer le voisinage :

```
09   N=neighbourhood(m,w,k,l)*ones(1,m{did});
10   ak=alpha(k,l);
```

Recalculer les positions des centroïdes avec (3.4) :

```
11   md=md+ak*N.*xk_array;
```

Pour les cartes auto-organisatrices à produit scalaire, il faut avoir des entrées et centroïdes normalisés.

```
12   md=md./((sum(md.^2,2)/m{did}).^(1/2)*ones(1,m{did}));
13 end
14 m{lid}=md;
```

Le centroïde le plus proche de l'entrée actuelle est déterminé en cherchant le produit scalaire maximale entrée-centroïde.

```
15 function rownumber=winner(md,xk) dp=md*xk';
16 [maximal_dp,rownumber]=max(dp);
```

Il ne faut pas oublier l'évolution de la largeur du pas.

```
17 function out=alpha(k,l)
```

Initialement, la largeur diminue vite :

```
18 if k<l,
19   out=1-k/(2*l);
```

Intermédiairement, la diminution ralentit :

```
20 elseif k<5*l,
21   out=.5*(1-(k-l)/(4*l));
```

Dans la phase finale, on garde la largeur constante :

```
22 else,
23   out=1e-4;
24 end
```

B.2 Modèles de régression

Les programmes montrées ici sont l'implémentation de la théorie du chapitre 4.

Déterminer les paramètres d'un modèle linéaire avec l'erreur quadratique moyenne comme critère d'erreur peut être facilement programmé avec la fonction de MATLAB® pour résoudre des systèmes au sens des moindres carrés.

AR.m

```
01 function parameters=AR(regressor,topredict)
```

Ici, `regressor` est une matrice avec les régresseurs comme lignes et `topredict` contient les grandeurs qu'on veut prédire.

```
02 parameters=regressor\topredict;
```

Pour pouvoir comparer les résultats, on a besoin de l'erreur de validation (voir §4.2.4). On la calcule avec la méthode basée sur le bootstrap (§4.2.5).

normedtotalerror.m

```
01 function nte=normedtotalerror(regressor,topredict,...
                                  bootindices)
```

Les `bootindices` sont des permutations des indices des lignes de `regressor` et de `topredict`.

```
02 parameters=AR(regressor,topredict);
```

D'abord, on calcule l'erreur d'apprentissage :

```
03 err=ARerror(regressor,topredict,parameters);
```

Le terme de correction :

```
04 booterr=booterror(regressor,topredict,bootindices);
```

L'erreur de validation qui en découle par l'équation (4.3) :

```
05 nte=(err+booterr);
```

L'erreur d'apprentissage qu'on utilise est l'erreur quadratique moyenne normalisé de l'équation (4.2).

ARerror.m

```
01 function error=ARerror(regressor,topredict,parameters)
02 n=normalizationfactor(topredict);
03 error=sum((topredict-regressor*parameters).^2)/n;
```

normalizationfactor.m

```
01 function n=normalizationfactor(topredict)
02 n=sum((topredict-mean(topredict)).^2);
```

Finalement, le terme de correction est calculé comme suit.

booterror.m

```
01 function booterror=booterror(regressor,topredict,...
                                bootindices)
02 b=size(bootindices,2);
03 e=0;
04 be=0;
05 for i=1:b,
06   btopredict=topredict(bootindices(:,i));
07   bregressor=regressor(bootindices(:,i),:);
08   bparameters=AR(bregressor,btopredict);
09   e=e+ARerror(regressor,topredict,bparameters);
10   be=be+ARerror(bregressor,btopredict,bparameters);
11 end
12 booterror=(e-be)/b;
```

Les calculs pour le modèle non linéaire qu'on a utilisé (basé sur le réseaux à RBF) sont faits avec le programme «RBFm» de J.A. Lee (regardez [22]).

Bibliographie

[1] G. Bastin and L. Moens. Hydromax: A Real-Time Application for River Flow Forecasting. In F. Toensmann and M. Koch, editors, *River Flood Defence*, volume 9 of *Kassel Reports of Hydraulic Engineering*, pages D33–D40, Kassel, Germany, 2000. Herkules, Kassel, Germany.

[2] A. Lendasse, M. Cottrel, V. Wertz, and M. Verleysen. Prediction of Electric Load using Kohonen Maps - Application to the Polish Electricity Consumption. In *Proc. of ACC 2002, the 2002 American Control Conference*, pages 3684–3689, Anchorage, Alaska, 2002.

[3] T. Kohonen. *Self-Organising Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, second edition, 1997.

[4] A.S. Weigend and N.A. Gershenfeld. The Future of Time Series: Learning and Understanding. In A.S. Weigend and N.A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume XV of *Studies in the Sciences of Complexity*, pages 1–70, Santa Fe, New Mexico, 1993. Addison-Wesley, Reading, Massachusetts.

[5] B. Wéry. *Identification des systèmes hydrologiques: application à la prévision des crues*. Thèse pour l'obtention du grade de Docteur en Sciences Agronomiques, Université Catholique de Louvain, Louvain-la-Neuve, 1990.

[6] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[7] D.O. Hebb. *The Organization of Behavior: A neuropsychological theory*. Wiley, New York, 1949.

[8] B. Widrow and M.E. Hoff. Adaptive switching circuits. In *1960 IRE Western Electric Show and Convention Record, Part 4*, pages 96–104, 1960.

[9] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:368–408, 1958.

[10] F. Blayo et M. Verleysen. *Les réseaux de neurones artificiels*. Que sais-je n^o 3042. Presses universitaires de France, Paris, première edition, 1996.

[11] M. Cottrell, B. Girard, and P. Rousset. Long Term Forecasting by Combining Kohonen Algorithm and Standard Prevision. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicard, editors, *Proc. of ICANN'97, 7th International Conference on Artificial Neural Networks*, volume 1327 of *Lecture Notes in Computer Science*, pages 993–998, Lausanne, Switzerland, 1997. Springer-Verlag, Berlin.

[12] M. Cottrell and P. Rousset. The Kohonen Algorithm: a Powerful Tool for Analysing and Representing Multidimensional Quantitative an Qualitative Data. In J. Mira, R. Moreno-Diaz, and J. Cabestany, editors, *Proc. of IWANN'97, International Work Conference on Artificial and Neural Networks, Biological and Artificial Computation: from Neuroscience to Technology*, volume 1240 of *Lecture Notes in Computer Science*, pages 861–871, Lanzarote, Canary Islands, Spain, 1997. Springer-Verlag, Berlin.

[13] M. Cottrel, B. Girard, and P. Rousset. Forecasting of Curves using a Kohonen Classification. *Journal of Forecasting*, 17(5-6):429–439, 1998.

[14] P. Rousset. Curve Forecasting with SOM Algorithm: Using a Tool to follow the Time on a Kohonen Map. In M. Verleysen, editor, *Proc. of ESANN'2000, 8th European Symposium on Artificial Neural Networks*, pages 353–358, Bruges, Belgium, 2000. D-Facto, Evere, Belgium.

[15] N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen. Width optimization of the gaussian kernels in radial basis function networks. In M. Verleysen, editor, *Proc. of ESANN'2002, 10th European Symposium on Artificial Neural Networks*, pages 425–432, Bruges, Belgium, 2002. D-Facto, Evere, Belgium.

[16] M.J.L. Orr. Introduction to Radial Basis Function Networks. `http://www.anc.ed.ac.uk/~mjo/papers/intro.ps`, April 1996.

[17] T. Poggio and F. Girosi. Networks for approximation and learning. In *Proc. of the IEEE*, volume 78, pages 1481–1497, 1990.

[18] M. Verleysen and K. Hlavackova. Learning in RBF Networks. In *Proc. of ICNN'96, International Conference on Neural Networks*, pages 199–204, Washington, DC, 1996. IEEE Service Center, Piscataway, New Jersey.

[19] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, New York, first edition, 1993.

[20] L. Ljung and T. Glad. *Modelling of Dynamic Systems*. Prentice-Hall information and system sciences series. Prentice-Hall, Englewood Cliffs, New Jersey, first edition, 1994.

[21] P. Rousset. *Application des algorithmes d'auto-organisation à la classification et à la prévision*. Thèse pour obtenir le grade de docteur de l'Université Paris 1, Université Paris 1 – Pantheon Sorbonne, Paris, 1999.

[22] J.A. Lee. RBFm. `http://www.dice.ucl.ac.be/~lee/software/rbf/main.html`, 2002.