# Apprentissage actif pour l' approximation de variétés

Benoît Gandar



N° d'ordre : 2293


**Université Blaise Pascal - Clermont II**

École Doctorale SPI

*Sciences Pour l'Ingénieur* N° 584


# THÈSE

présentée par


## Benoît GANDAR


pour obtenir le grade de

## Docteur d'Université

Spécialité : Informatique

# APPRENTISSAGE ACTIF POUR L'APPROXIMATION DE VARIÉTÉS

Soutenue publiquement le 27 novembre 2012 devant le jury composé de :

| | | |
|---|---|---|
| M. Philippe MAHEY | Professeur des Universités, Clermont-Ferrand | Président |
| M. Stéphane CANU | Professeur des Universités, INSA Rouen | Rapporteur |
| M. Antoine CORNUÉJOLS | Professeur des Universités, AgroParisTech | Rapporteur |
| M. Olivier TEYTAUD | Chargé de Recherche, HDR, INRIA Saclay | Examinateur |
| M. Guillaume DEFFUANT | Directeur de Recherche, Irstea | Directeur de thèse |
| M^lle Gaëlle LOOSLI | Maître de Conférences, Clermont-Ferrand | Encadrante de thèse |


Équipes d'accueil :
- Laboratoire d'Ingénierie pour les Systèmes Complexes
  Irstea de Clermont-Ferrand
- Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
  Université Blaise Pascal

« Soit $A$ un succès dans la vie.
Alors $A = x + y + z$,
où x=travailler, y = s'amuser, z=se taire. »
ALBERT EINSTEIN.
Extrait de *The observator*, 1954.

« Dans les champs de l'observation,
le hasard ne favorise que les esprits préparés. »
LOUIS PASTEUR.
Discours prononcé à Douai, le 7 décembre 1854.

« Ce qui est incompréhensible,
c'est que le monde soit compréhensible. »
ALBERT EINSTEIN.
Extrait de *Comment je vois le monde* , 1934.

# AVANT-PROPOS

CETTE THÈSE a été réalisée au Laboratoire d'Ingénierie pour les Systèmes Complexes (LISC) [1] du Cemagref de Clermont-Ferrand. Depuis décembre 2011, le Cemagref est devenu l'Irstea [2], l'Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture.

Cette thèse a bénéficié de fortes collaborations avec le Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS) [3] de l'Université Blaise Pascal.

Cette thèse a pu être réalisée grâce au soutien financier de la région Auvergne [4] dans le cadre de sa politique de participation au développement de la recherche, notamment à travers le financement d'allocations de recherche.


1. http://motive.cemagref.fr/lisc
2. www.irstea.fr
3. http://limos.isima.fr/
4. www.auvergne.org

## - RÉSUMÉ -

L'apprentissage statistique cherche à modéliser un lien fonctionnel entre deux variables $X$ et $Y$ à partir d'un échantillon aléatoire de réalisations de $(X,Y)$. Lorsque la variable $Y$ prend un nombre binaire de valeurs, l'apprentissage s'appelle la classification (ou discrimination en français) et apprendre le lien fonctionnel s'apparente à apprendre la frontière d'une variété dans l'espace de la variable $X$.

Dans cette thèse, nous nous plaçons dans le contexte de l'apprentissage actif, *i.e.* nous supposons que l'échantillon d'apprentissage n'est plus aléatoire et que nous pouvons, par l'intermédiaire d'un oracle, générer les points sur lesquels l'apprentissage de la variété va s'effectuer.

Dans le cas où la variable $Y$ est continue (régression), des travaux précédents montrent que le critère de la faible discrépance pour générer les premiers points d'apprentissage est adéquat. Nous montrons, de manière surprenante, que ces résultats ne peuvent pas être transférés à la classification. Dans ce manuscrit, nous proposons alors le critère de la dispersion pour la classification. Ce critère étant difficile à mettre en pratique, nous proposons un nouvel algorithme pour générer un plan d'expérience à faible dispersion dans le carré unité.

Après une première approximation de la variété, des approximations successives peuvent être réalisées afin d'affiner la connaissance de celle-ci. Deux méthodes d'échantillonnage sont alors envisageables : le « selective sampling » qui choisit les points à présenter à un oracle parmi un ensemble fini de candidats et l'« adaptative sampling » qui permet de choisir n'importe quels points de l'espace de la variable $X$. Le deuxième échantillonnage peut être vu comme un passage à la limite du premier. Néanmoins, en pratique, il n'est pas raisonnable d'utiliser cette méthode. Nous proposons alors un nouvel algorithme basé sur le critère de dispersion, menant de front exploitation et exploration, pour approximer une variété.



## - MOTS CLEFS -

Apprentissage statistique, apprentissage actif, échantillonnage aveugle ou échantillonnage adaptatif, échantillonnage sélectif, plans d'expériences, approximation de variétés, discrépance, dispersion, maximin, minimax.

## - ABSTRACT -

Statistical learning aims to modelize a functional link between two variables $X$ and $Y$ thanks to a random sample of realizations of the couple $(X,Y)$. When the variable $Y$ takes a binary number of values, learning is named classification and learn the functional link is equivalent to learn the boundary of a manifold in the feature space of the variable $X$.

In this PhD thesis, we are placed in the context of active learning, *i.e.* we suppose that learning sample is not random and that we can, thanks to an oracle, generate points for learning the manifold.

In the case where the variable $Y$ is continue (regression), previous works show that criterion of low discrepacy to generate learning points is adequat. We show that, surprisingly, this result cannot be transfered to classification talks. In this PhD thesis, we propose the criterion of dispersion for classification problems. This criterion being difficult to realize, we propose a new algorithm to generate low dispersion samples in the unit cube.

After a first approximation of the manifold, successive approximations can be realized in order to refine its knowledge. Two methods of sampling are possible : the « selective sampling » which selects points to present to the oracle in a finite set of candidate points, and the « adaptative sampling » which allows to select any point in the feature space of the variable $X$. The second sampling can be viewed as the infinite limit of the first. Nevertheless, in practice, it is not reasonable to use this method. Then, we propose a new algorithm, based on dispersion criterion, leading both exploration and exploitation to approximate a manifold.



## - KEY WORDS -

Statistical learning, active learning, blind active learning, selective active learning, experimental design, manifolds approximation, discrepancy, dispersion, maximin, minimax.

# REMERCIEMENTS

En premier lieu, je tiens à remercier les personnes qui m'ont fait l'honneur de participer à mon jury. Je remercie Stéphane Canu et Antoine Cornuéjols pour leur intérêt à rapporter cette thèse en y apportant chacun un éclairage différent. Je remercie Philippe Mahey pour m'avoir fait l'honneur de présider ce jury. Je remercie Olivier Teytaud pour les échanges que nous avons eus et pour avoir accepté de faire partie du jury.

Je souhaite ensuite remercier très chaleureusement Guillaume pour m'avoir fait confiance dès notre première rencontre et pour m'avoir ensuite encadré dans cette thèse. Merci pour tes idées, tes relectures, ton écoute, ton soutien, tes encouragements et tes conseils.
Je remercie aussi très chaleureusement Gaëlle pour toutes ces années d'« apprentissage », pour le débuggage de codes, pour les relectures, les conseils et nos nombreuses discussions. Tu es un peu ma « grande sœur » de la recherche ! ;-)

Je souhaite remercier Philippe Mahey pour m'avoir confié des enseignements à l'Isima. Merci à Claude Mazel et à Vincent Barra pour leur aide dans la préparation des cours.

Je souhaite remercier la région Auvergne qui a financé ces travaux de thèse.

Ces années de thèse n'auraient pas été les mêmes sans l'entourage du laboratoire. Les pauses café (ou plutôt thé) ont réellement contribué à la bonne ambiance générale : à titre personnel, elles m'ont aidé à faire face aux difficultés de la thèse grâce à toutes ces discussions plus ou moins scientifiques...
Je garderai un très bon souvenir du premier « Courir à Clermont » (et de sa préparation avec Laëtitia), des parties de pétanque (Tom & $C^{ie}$), des déjeuners croque-monsieurs ou raclettes qui embaumaient tout le couloir, des footings (Franck & $C^{ie}$),...
Merci à vous, Claire, Thomas, Nabil, Wei, Franck, Jean-Denis, Bruno, Sylvie, Guillaume, Thierry, Laëtitia, Clarisse, Nicolas, Charles, Isabelle, Sophie, Maxime ... pour tous ces bon moments. Rendez-vous quand vous voulez pour un footing ou pour partager un verre de rosé ou de Viognier !

Merci aux personnes qui ont partagé mon bureau, et qui m'ont supporté : Laëtitia, Claire, Thomas et Gaëlle.
Merci à Clarisse pour son aide administrative et pour son amitié.

Merci aux relecteurs de ce manuscrit qui se reconnaîtront !

Je remercie mes nouveaux collègues de Bibendum pour m'avoir attendu et m'avoir laissé le temps de finir ce manuscrit.

Je souhaite remercier sincérement les cigales ardéchoises qui, de leur chant, m'ont accompagné et égayé durant la rédaction de ce manuscrit !
J'ai également une pensée pour toutes les bananes de la pause de 16h00. Ainsi qu'aux macarons que nous dégustions de manière Joyeuse ( :-)) entre initiés, une fois la porte fermée...

Je remercie également mes amis qui ont toujours été là pour me changer les idées. Merci aux Clermontois : Wei (promis, je vais essayer de manger avec les baguettes), Tom (j'ai toujours des glaçons, du pastis et des olives), Clairus. Merci à Nabil (on remet quand les baskets ? et le couscous ?). Merci à Gaëlle. Merci à Nico pour nos discussions téléphoniques hautement philosophiques, pour les vacances, pour les voyages, pour toute notre amitié et pour tout le reste ! Merci à l'ensemble des Zozos pour ce que l'on est !

Mes dernières remerciements vont vers ma famille qui m'a toujours soutenu et encouragé. Pour le vin, pour les corridas, pour les bons moments, pour leurs plaisanteries (spéciales dédicaces à mes soeurs et $C^{ie}$...) et pour tout le reste...

Enfin, je conclurai en remerciant tous ceux que j'ai oubliés, en leur adjoignant mes excuses pour cet oubli...

# TABLE DES MATIÈRES

# NOTATIONS

## Mathématiques générales

$\mathbb{R}$ ..... Espace des réels
$\mathbb{N}$ ..... Espace des entiers
$\mathbb{N}^*$ ..... Espace des entiers non nuls
$s$ ..... Dimension de l'espace
$\lambda$ ..... Mesure de Lebesgue
$I^s = [0,1]^s$ ..... Cube unité en dimension $s$

$\#(X)$ ou $\mathscr{C}ard(X)$ ..... Cardinal de l'ensemble X
$\lceil x \rceil$ ..... Partie entière supérieure de $x$
$\lfloor x \rfloor$ ..... Partie entière inférieure de $x$
$\arg\max_x S(x)$ ..... Sélection de la valeur de x qui donne le maximum pour l'expression S
$\arg\min_x S(x)$ ..... Sélection de la valeur de x qui donne le minimum pour l'expression S
$\mathscr{O}$ ..... Ordre de grandeur maximal de complexité d'un algorithme

$x_n$ ..... $n^{\text{ième}}$ élément d'une suite
$x_{(n)}$ ..... Suite de $n$ éléments
$x_{(n)} = \{x_1, \ldots, x_n\}$
$x_n^i$ ..... $i^{\text{ième}}$ coordonnée du $n^{\text{ième}}$ élément de la suite $x$

$B(x,R)$ ..... Boule de centre $x$ et de rayon $R$
$\mathbb{1}_A(x) = \begin{cases} 1 & \text{si} \quad x \in A \\ 0 & \text{si} \quad x \notin A \end{cases}$ ..... Fonction indicatrice

$d(x,y)$ ..... Distance euclidienne entre les points $x$ et $y$
$D(x)$ ..... Discrépance extrême de la suite $x$
$D^*(x)$ ..... Discrépance à l'origine de la suite $x$
$J(x)$ ..... Discrépance isotrope de la suite $x$
$\delta(x)$ ..... Dispersion de la suite $x$
$V_{HK}(f)$ ..... Variation au sens d'Hardy-Krause d'une fonction $f$

## Probabilité - Statistiques

$\mathscr{P}$ ..... Loi de probabilités
$\mathbb{P}$ ..... Mesure de probabilité selon une certaine loi
$\mathbb{E}$ ..... Espérance mathématique
$\mathbb{V}ar$ ..... Variance mathématique

## Apprentissage

$\mathscr{A}$ ou $\mathscr{M}$ ..... Algorithme ou Modèle d'apprentissage
$\mathscr{X}$ ..... Espace des instances
$\mathscr{Y}$ ..... Espace des étiquettes
$D_n$ ..... Ensemble de $n$ exemples étiquetés

$\mathbb{H}$ ..... Espace d'hypothèses
$\mathscr{VC}$ ..... Dimension de Vapnik-Chervonenkis

$l$ ..... Fonction de coût
$\mathscr{R}[f]$ ..... Fonction de risque de la fonction $f$
$\mathscr{R}^* = \inf_{f \in \mathbb{H}} \mathscr{R}[f]$ ..... Risque minimum atteint sur l'espace $\mathbb{H}$
$\hat{\mathscr{R}}_n[f]$ ou $\hat{\mathscr{R}}[f]$ ..... Fonction de risque empirique de la fonction $f$

$f$ ..... Fonction cible
$\hat{f}$ ..... Fonction appartenant à l'espace d'hypothèses $\mathbb{H}$
$f^*_{\mathbb{H}} = \arg\min_{g \in \mathbb{H}} \mathscr{R}[g]$ ..... Fonction de $\mathbb{H}$ qui minimise le risque en généralisation
$\hat{f}_n = \arg\min_{g \in \mathbb{H}} \hat{\mathscr{R}}_n[g]$ ..... Fonction de $\mathbb{H}$ qui minimise le risque empirique
$f^*$ ..... Fonction de Bayes

## Sigles

Irstea ..... Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture
LISC ..... Laboratoire d'Ingénierie pour les Systèmes Complexes (Irstea)
LIMOS ..... Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes

*k*NN ..... Algorithme des $k$ plus proches voisins
MCMC ..... Méthode de Monte-Carlo par Chaîne de Markov
MH ..... Métropolis-Hastings (algorithme de)
PAC ..... Probablement Approximativement Correct
RSCM ..... Recuit Simulé sur le Critère du *Minimax* (algorithme de)
SGCM ..... Sous-grille cubique minimale
SVM ..... Support Vector Machine, Séparateur à Vaste Marge

# INTRODUCTION

## Contexte

### De la modélisation des systèmes complexes à l'approximation de variétés

Au cours de ces dernières décennies, la modélisation a pris une place de plus en plus importante dans les travaux de recherche des scientifiques, toutes disciplines confondues. Le développement et l'automatisation des méthodes permettant de récolter (ou de simuler) les données expérimentales nécessaires à toute modélisation, ainsi que l'intégration de nouvelles connaissances aux modèles déjà existants, permettent de concevoir des modèles de plus en plus précis. Cependant, ce développement, couplé aux progrès des méthodes mathématiques et surtout au développement des capacités informatiques à réaliser et à stocker des calculs, rendent aujourd'hui ces modèles de plus en plus **complexes** [5].

L'étude de ces modèles complexes représente alors un véritable enjeu pour comprendre les phénomènes étudiés. Lors de ces études ou de l'étude d'un modèle couplant plusieurs modèles différents, nous cherchons souvent à déterminer (identifier) certaines zones de paramètres en fonction de leurs différentes propriétés.

Lors de la conception du modèle, il est souvent nécessaire de réaliser une étape de calage (ou de calibrage) de celui-ci. Par exemple, en ingénierie nucléaire, Kennedy et O'Hagan (2001) doivent estimer le taux d'émission de radionucléides (ou radioisotopes) pour des modèles d'accidents nucléaires. En hydrologie, Romanowicz (2006) doit estimer des paramètres liés à la transmissivité hydraulique dans des modèles de pluie. Pour cela, nous cherchons alors la zone de paramètres pour lesquels le modèle présente une erreur (calculée à l'aide d'un échantillon de données) inférieure à un certain seuil. Cela revient à cartographier l'espace des paramètres en une variété à deux zones. La première zone correspond à l'ensemble des paramètres où le modèle réalise une erreur faible (inférieure à un certain seuil) et la deuxième zone à l'ensemble des paramètres où le modèle réalise une erreur importante.

Dans l'étude de modèles dynamiques, nous pouvons chercher les zones de paramètres dans lesquelles le modèle a un certain type de régime dynamique. Il s'agit, là aussi, de chercher une variété séparant l'espace des paramètres en zones, en fonction du type de régime dynamique du modèle ayant ses paramètres présents dans la zone. Par exemple, dans un modèle d'écologie, le problème est de maintenir la pérennité d'une ressource renouvelable. Dans ce cas, on cherche à maintenir la population au-dessus d'une certaine valeur pour laquelle l'extinction est inévitable. Dans le domaine de la gestion d'une ressource marine, Delara *et al.* (2007) analysent la pérennité

5. Nous entendons par complexe un modèle qui est constitué d'un grand nombre de paramètres et/ou de variables en interaction qui empêchent l'observateur de prévoir son comportement ou son évolution par le calcul.

d'un écosystème constitué de merlus et d'anchois dans le golfe de Gascogne, en fonction du niveau de pêche et du recrutement des espèces (nombre de jeunes poissons constituant la nouvelle classe d'âge annuelle). Ils ont identifié les configurations qui permettent de conserver la durabilité du système, c'est-à-dire de maintenir la population de chaque espèce au-dessus de certaines valeurs.

Dans un registre similaire, il est fréquent de vouloir cartographier le comportement du modèle en fonction de ses conditions initiales. Ce type d'étude permet en effet d'obtenir une vision plus globale du comportement du modèle vis-à-vis de la propriété considérée. Un des buts de cette étude peut être de détecter le nombre d'(états) attracteurs du modèle. Par exemple, Bonneuil (2003) étudie les conditions que doivent respecter les dynamiques d'un système proie-prédateur afin d'éviter l'extinction de l'une ou l'autre espèce.
Une autre application, utilisée au LISC [6], concerne le contrôle des systèmes en utilisant la théorie de la viabilité définie par Aubin (1991). Cette théorie propose des concepts et méthodes pour contrôler un système dynamique non stochastique afin de le maintenir dans un ensemble de contraintes. A partir du calcul du noyau de viabilité du système, elle permet de définir des politiques d'action qui maintiennent le système dans un ensemble de contraintes choisi. Les valeurs de ces actions réalisées sur le système constituent alors les variables du problème et on cherche à cartographier la réponse du système à ces actions. Dans le cadre de la théorie de la variété, les réponses se classent en deux catégories (état viable du système ou état non viable). Cela revient donc à déterminer les **frontières d'une surface ou d'une variété** ne comprenant que deux valeurs possibles. Un algorithme d'approximation de cette surface, développé par Saint-Pierre (1994) [7], permet d'approcher cette variété par approximations successives.

**Notre problème consiste donc à approcher la frontière d'une zone continue de l'espace. Cette zone est approchable par une variété continuement différentiable. Cette dernière peut alors être approchée par une variété continue. Notre problème se ramène donc à s'approcher d'une variété continue de l'espace, autrement dit d'un objet géométrique obtenu par recollement d'ouverts dans l'espace des variables.**

## De l'étude des modèles complexes et des variétés à l'apprentissage

L'étude de ces modèles, lorsqu'on se situe dans des espaces de dimension élevée, pose cependant des difficultés particulières. En effet, il devient difficile de visualiser les résultats dans l'espace. De plus, pour obtenir des approximations précises des surfaces de réponse, il est nécessaire de tester un nombre de points qui croît exponentiellement avec la dimension de l'espace : c'est la malédiction de la **dimensionnalité** ! Non seulement ces travaux réclament une puissance de calcul et un espace mémoire souvent hors de portée, de plus les résultats produits sont difficiles à interpréter et à utiliser.

Pour lever, au moins partiellement, ces difficultés, il est d'usage d'approximer ces surfaces de réponse par **apprentissage statistique** à partir des données résultant d'expériences ou de simulations. En effet, le principe de l'apprentissage est de représenter de manière la plus concise possible, par l'intermédiaire de fonctions mathématiques appropriées, certaines propriétés d'échantillons de points dans l'espace. Les surfaces de réponse sont ainsi représentées par des objets de taille plus raisonnable : le **méta-modèle**. De plus, ces fonctions permettent de réaliser des prédictions du modèle sans avoir à exécuter de simulations, ce qui peut être un gain de temps très appréciable. Ceci facilite alors l'étude des propriétés de cette surface de réponse (par exemple les propriétés topologiques, et leur stabilité éventuelle). A titre illustratif, Jourdan et Zabalza-Mezghani (2004) utilisent une méthode de régression pour optimiser un méta-modèle d'incertitude de réservoir en ingénierie géologique et Chapel et Deffuant (2007) utilisent la méthode des SVMs pour déterminer la surface

6. Laboratoire d'Ingénierie des Systèmes Complexes, Irstea de Clermont-Ferrand.
7. Cet algorithme n'est pas considéré comme un algorithme d'apprentissage pour bien le situer dans la logique de ce chapitre.

de réponse du modèle et calculer à partir de cette dernière les politiques d'actions viables du système.

Le problème de la construction de ces surfaces de réponse, et donc de ces modèles, revient donc à un problème d'apprentissage statistique. Plus formellement, les données, sur lesquelles sont construits ces méta-modèles, sont constituées d'instances $x \in \mathscr{X} \subset \mathbb{R}^s$ (les variables) et de la réponse du modèle $y \in \mathscr{Y} \subset \mathbb{R}$ en ces données d'observations. Une observation est donc constituée du couple $(x,y)$. Grâce à plusieurs observations, l'apprentissage statistique permet alors de trouver un lien fonctionnel $f$ entre $x$ et $y$ de telle façon que $y = f(x)$.
Dans le cadre de l'étude de modèles, les réponses possibles des méta-modèles sont la plupart du temps en nombre fini, comme présenté dans les paragraphes précédents. Par exemple, la sortie du méta-modèle possède telle propriété ou non. Ceci implique que l'ensemble $\mathscr{Y}$ des valeurs prises par le modèle est un ensemble discret de cardinalité finie ; les valeurs de $x$ délivrant la même réponse ou la même propriété du méta-modèle forment alors une même classe, et l'ensemble des classes forment une **variété**. Ce type d'apprentissage de variétés s'appelle la **classification** et nous nous restreindrons alors à ce cas d'étude. **Nous attirons l'attention du lecteur sur le sens du mot « classification » utilisé dans cette thèse : nous utilisons le mot classification avec son sens anglais ; sa traduction dans la langue de Molière serait plutôt discrimination.**

Dans le cadre de l'apprentissage statistique, les données sont issues d'observations provenant de réalisations de variables aléatoires. Dans le cadre de l'étude de ces modèles ou de ces variétés, une particularité existe néanmoins : le modélisateur interagit avec le modèle et peut, soit en allant directement récolter les données sur le terrain, soit par le biais de simulations informatiques, choisir directement les données sur lesquelles va être construit le méta-modèle. L'apprentissage n'est alors plus considéré comme statistique. Tout comme l'apprentissage statistique, cette stratégie exploite alors les données potentiellement disponibles ainsi qu'un algorithme d'apprentissage pour inculquer un comportement à un modèle prédictif. En choisissant convenablement les données à utiliser pour l'apprentissage, on peut aisément imaginer que le modèle réalisé sera de qualité prédictive supérieure, et permettra ainsi de repousser encore temporairement cette malédiction de la dimensionnalité. On parle alors d'**apprentissage actif**. Le problème initial d'étude des modèles revient donc à un problème d'**apprentissage actif.**

## La problématique de l'apprentissage actif

Une des problématiques de l'apprentissage actif concerne donc le choix de l'échantillon des données que l'on va utiliser pour réaliser le méta-modèle ou apprendre la variété.
Dans la suite de cette introduction et de cette thèse, nous n'utilisons plus le terme « modèle » au sens large du terme. Nous réservons ce terme au cas de la modélisation au sens de l'apprentissage, *i.e.* à partir d'observations $(x,y)$ apprendre une fonction $f$ qui établisse un lien entre $x$ et $y$ de telle façon que $f(x) = y$. Castro *et al.* (2005) distinguent deux familles de scénarii possibles pour l'apprentissage actif : l'**échantillonnage sélectif** et l'**échantillonnage adaptatif**. Dans le cadre de l'apprentissage sélectif, une multitude d'instances (exemples non étiquetés) est disponible. Cet ensemble d'instances est cependant de cardinalité finie. Le problème revient alors à choisir les instances les plus pertinentes pour l'apprentissage. Une fois choisies, elles sont présentées à un expert ou un oracle qui les étiquette, puis on les rajoute à la base d'exemples initiale pour réaliser l'apprentissage. Cette stratégie ne peut donc observer qu'une partie restreinte de l'espace des entrées, matérialisée par les exemples. Dans le cadre de l'échantillonnage adaptatif, la stratégie est différente. Tout point de l'espace des entrées peut être techniquement étiqueté par l'expert ou l'oracle. La problématique revient donc à générer les points d'apprentissage les plus adéquats. Cela revient à sélectionner des instances dans un ensemble dense. L'échantillonnage adaptatif peut être vu comme un passage à la limite de l'échantillonnage sélectif. Théoriquement, cela est possible : il suffit de générer une grille de pas régulier sur l'espace des entrées. En faisant tendre le pas vers zéro, le

nombre de points de la grille tend vers l'infini, et la grille couvre tout l'espace. Il ne reste plus qu'à sélectionner les points les plus « informatifs ». Néanmoins, en pratique cette approche n'est pas réalisable. En effet, pour permettre de sélectionner ou non chacune des instances candidates, elle requiert de nombreux calculs qui, le plus souvent, sont redondants. Par exemple des points voisins sur la précédente grille et qui se situent loin des frontières de la variété auront une influence similaire en cas d'ajout dans la base d'apprentissage, et les calculs engendrés pour les sélectionner ou non seront également proches et pas conséquent redondants. En pratique, il est donc nécessaire de distinguer les méthodes d'échantillonnage sélectif des méthodes d'échantillonnage adaptatif.

Nous pouvons également (voir Hoi *et al.* (2006) ou Sugiyama et Rubens (2008)) distinguer théoriquement deux autres familles de scenarii pour l'apprentissage actif : l'**apprentis- sage en ligne** et l'**apprentissage en mode** « **batch** ». Dans le cadre de l'apprentissage en ligne, on sélectionne ou génère un seul exemple à la fois. Suite à l'étiquetage de ce dernier, il est intégré à la base d'exemples étiquetés, et le problème est alors mis à jour en calculant une nouvelle solution prenant en compte cette nouvelle donnée. Cependant, lorsque le modèle ne peut pas être modifié localement et qu'il est nécessaire de le recalculer entièrement, ou lorsque la réalisation d'un modèle demande une énergie importante (temps de calcul, espace mémoire,...), cette stratégie n'est pas réalisable. Nous réalisons alors un apprentissage « batch », que l'on pourrait traduire en français par apprentissage par « lot ». Cela consiste à sélectionner ou générer en même temps plusieurs points d'apprentissage et à les étiqueter. Puis, ces points sont rajoutés dans la base d'apprentissage, et on réapprend le modèle sur cette dernière. Le nombre de points à rajouter à chaque étape est défini par l'utilisateur. Cette stratégie est surtout efficace lorsque l'étiquetage de points est rapide, qu'il peut être, par exemple, réalisé en parallèle, et que le calcul du modèle est long ou coûteux.

Enfin, deux autres types d'actions peuvent être encore distingués. Une action où l'on cherche à **explorer** le modèle et une action où l'on cherche plutôt à **exploiter** celui-ci. L'apprentissage consiste alors à réaliser un compromis efficace entre exploration et exploitation. Ceci s'illustre particulièrement dans le cadre de la classification. La partie exploration du modèle consiste alors à explorer les zones de l'espace des exemples où il n'y a pas ou peu d'exemples, et donc d'information. Cela permet par exemple de détecter de nouvelles zones de l'espace où différentes composantes de la variété existent. La phase d'exploration permet, quant à elle, d'augmenter localement la précision que l'on a du modèle (ou de la variété). Cela permet de réduire les zones d'incertitude autour de la frontière de la variété et donc d'augmenter la qualité du modèle. Ces deux stratégies peuvent être menées de front, cependant elles sont opposées. En effet, l'exploration permet d'avoir une connaissance globale du modèle sur son ensemble de définition alors que l'exploitation permet de raffiner localement sa connaissance.

Ces trois types de stratégie sont différents mais pas pour autant contradictoires. Les deux premiers, échantillonnage sélectif ou adaptatif et apprentissage en ligne ou en mode batch, sont, la plupart du temps, subis par l'apprenant : en effet, cela dépend des conditions extérieures imposées à l'apprenant pour obtenir de nouvelles données. Le dernier type, exploration / exploitation, dépend des objectifs à atteindre. Une stratégie d'apprentissage actif qui ne fait que de l'exploration ne se focalise pas sur des zones de l'espace de données où le modèle aurait besoin d'être amélioré. La stratégie d'apprentissage actif présente alors peu d'intêret par rapport à un apprentissage statistique où les données sont aléatoires. Au contraire, une stratégie d'apprentissage actif qui ne fait qu'exploiter le modèle court le risque d'occulter une partie de l'espace des observations pour générer les nouvelles données. Le modèle sera alors très spécialisé dans certaines zones de l'espace des observations, mais aura une capacité de généralisation mauvaise. Il est donc nécessaire de trouver un compromis à ce dilemme exploration/exploitation...

Le problème à résoudre est alors : **Quelle stratégie devons-nous utiliser pour générer les données d'apprentissage afin d'augmenter la qualité de l'apprentissage d'une surface de réponse**

**en classification, ou d'une variété ?**
Dans ce problème, nous pouvons distinguer deux configurations : une première configuration où nous n'avons aucun point d'apprentissage ainsi qu'aucune approximation de la variété ; et une deuxième configuration où une première approximation de la variété est déjà réalisée. Dans la première configuration, l'objectif est d'explorer l'espace des variables et donc de générer les premiers points d'apprentissage pour obtenir une première approximation correcte. Dans la deuxième configuration, il s'agit d'affiner itérativement l'approximation de la variété : il faut à la fois affiner la frontière apprise de la variété, et, en même temps, vérifier que toutes les composantes de la variété sont détectées. Il s'agit alors de trouver un bon compromis entre le dilemme exploration/exploitation...

## Apports de la thèse et organisation du mémoire

Dans cette thèse, nous étudions le problème de l'apprentissage actif et nous nous focalisons principalement sur le problème de la classification.

Le premier chapitre de la thèse commence par une présentation de la théorie de l'apprentissage statistique définie selon Vapnik (1995). Nous présentons la philosophie de l'apprentissage, les outils nécessaires à la compréhension et les principaux résultats de convergence. Nous définissons ensuite théoriquement l'apprentissage actif, distinguons ses différentes formes et présentons les principaux résultats théoriques. La distinction réalisée ainsi que la présentation ne sont pas exhaustives et ne sont fournies que dans le but de situer notre problématique et nos travaux de recherche convenablement dans l'état de l'art.

Dans la suite de la thèse, nous nous plaçons exclusivement dans le cadre de l'apprentissage actif, *i.e.* nous supposons qu'il existe un oracle ou un expert capable d'étiqueter les instances qui lui sont présentées. Ces instances sont, soit issues d'un ensemble fini, soit issues d'un ensemble dense.

La première partie de la thèse se situe dans le cadre de la génération des premiers points d'apprentissage. Cette étape est très importante car c'est sur ces points qu'est construit le premier modèle qui servira de base pour les autres sélections / générations de points.
Dans le cas de l'échantillonnage sélectif des $n$ premiers points d'apprentissage sans connaissance *a priori*, il paraît évident que ces points doivent être répartis les plus « uniformément » possible. Avec ce critère d'« uniformité », le problème revient donc à sélectionner parmi les instances disponibles, un ensemble qui satisfasse au mieux ce critère. Cette étape peut être très calculatoire, mais par des méthodes de partitionnement de l'espace, cela reste réalisable dans de nombreux cas. Dans le cas de l'échantillonnage adaptatif, la génération des $n$ premiers points d'apprentissage consiste donc « seulement » à générer des points selon ce critère.
Le problème consiste donc à déterminer quel critère d'« uniformité » il faut utiliser : des points aléatoirement uniformes ? des points à espacement régulier (ou uniforme) ? des points à distribution anisotrope ?...

Le second chapitre de cette thèse présente les résultats existants sur la génération de points dans le cadre des problèmes de régression et les compare à ceux obtenus avec l'apprentissage statistique. Le critère d'« uniformité » adéquat proposé est le critère de discrépance des points. Celui-ci mesure l'« uniformité » des points en prenant en compte notamment leur isotropie. Ces résultats se basent sur la théorie de l'intégration et le théorème de Koksma-Hlawka.

Le troisième chapitre est consacré à la transposition des résultats précédents au cas de la classification. Nous montrons alors que, de manière surprenante, cette démarche n'est pas transférable théoriquement. Nous illustrons expérimentalement ce résultat.

Dans le quatrième chapitre, nous proposons un nouveau critère d'« uniformité » pour générer les points d'apprentissage en classification. En nous basant sur des travaux d'optimisation numérique, nous proposons comme critère de minimiser la dispersion. Après l'avoir définie, nous établissons un lien théorique entre qualité d'apprentissage et dispersion dans un contexte d'apprentissage particulier. Puis, nous montrons expérimentalement que ce résultat peut se généraliser à de nombreux autres contextes d'apprentissage.

Le cinquième chapitre présente les différentes méthodes présentes dans la littérature pour générer des suites à faible dispersion. Cependant celles-ci ne délivrent pas nécessairement la configuration optimale. Nous présentons alors un nouvel algorithme basé sur le principe du recuit-simulé qui, en pratique, fournit de bons résultats.

La deuxième partie de la thèse est consacrée à l'étude de l'exploration et de l'exploitation des modèles de classification après un ou plusieurs apprentissages.

Le sixième chapitre s'intéresse aux méthodes d'échantillonnage sélectif d'instances. Nous présentons notamment les méthodes de sélection par incertitude, de sélection par arbre des voisins, l'échantillonnage par « Query by Committe », par réduction de l'espace des versions, l'échantillonnage bayésien et l'échantillonnage par strate.

Le septième et dernier chapitre s'intéresse à l'échantillonnage adaptatif. L'échantillonnage adaptatif est vu, parfois, comme un passage à la limite de l'échantillonnage sélectif : il suffit alors de générer une grande base d'exemples non étiquetés et d'y réaliser la sélection. Ce point de vue est réaliste en théorie, mais, en pratique, il n'est pas réalisable. Il convient donc de développer des méthodes d'échantillonnage adaptatif en temps que tel. Dans le cadre de l'échantillonnage adaptatif en classification, peu de travaux existent. En reprenant les travaux présentés au chapitre 3 sur la caractérisation des fonctions de classification, nous présentons alors un algorithme d'exploration et exploitation simultanées de variétés en mode batch à échantillonnage adaptatif.

# - CHAPITRE 1 -

# APPRENTISSAGE STATISTIQUE ET APPRENTISSAGE ACTIF

## Sommaire



LA NOTION D'APPRENTISSAGE est claire et intuitive pour les humains ou les animaux : il s'agit d'une procédure cognitive complexe qui vise à acquérir ou à développer certaines facultés. Après avoir réalisé l'apprentissage d'une tâche, typiquement à partir d'exemples, l'individu sera alors capable de la réaliser de manière autonome. Une illustration classique en est l'apprentissage de la reconnaissance des chiffres et des lettres pour un enfant : on lui présente des exemples de lettres et de chiffres, écrits avec des écritures et des fontes différentes. À la fin de l'apprentissage, on attend de l'enfant qu'il soit capable de lire non seulement tous les chiffres et lettres de son livre de lecture, mais également tous les chiffres et lettres qu'il est susceptible de voir : en d'autres termes, on attend de lui qu'il ait une capacité de généralisation à partir des exemples qui lui ont été présentés, sans qu'il ne soit jamais nécessaire de lui fournir une description analytique et discursive de la forme et de la topologie des chiffres et des lettres.

La notion d'apprentissage pour les machines consiste également à utiliser des exemples ou des observations pour apprendre afin d'améliorer leurs performances. Elle poursuit exactement le même objectif : il s'agit de faire en sorte, à l'aide d'une procédure numérique programmée et exécutée sur un ordinateur, d'inférer un modèle d'un processus que l'on observe et sur lequel on peut effectuer des mesures, *i.e.* un ensemble d'équations qui décrivent le processus observé et qui permettent de faire des prédictions concernant le comportement de celui-ci. À cette fin, on fait l'hypothèse que le

processus peut être décrit avec la précision désirée par une ou plusieurs fonctions qui contiennent des paramètres, et l'on ajuste ces derniers pour que cette ou ces fonctions s'ajustent aux données.

Ce chapitre se consacre à la formalisation du problème d'apprentissage. La première section commence par décrire le cadre mathématique du problème d'apprentissage statistique ainsi que ses outils. Ce chapitre n'a pas vocation à présenter de manière exhaustive tous les outils et les résultats qui existent dans la littérature abondante : il ne présente que les résultats qui ont une utilité pour la suite de cette thèse. Dans une deuxième section, nous présentons des résultats dans le cadre de l'apprentissage statistique en classification. Dans la troisième section, nous distinguons l'apprentissage statistique de l'apprentissage actif et nous montrons comment le problème d'approximation de variétés s'articule autour de l'apprentissage actif. Enfin, dans une quatrième section, nous concluons en définissant clairement le cadre d'apprentissage dans lequel nous nous situons pour approximer des variétés.

## 1.1 Cadre mathématique de l'apprentissage statistique

Formulée par Vapnik (1995), l'apprentissage statistique se trouve au carrefour de différentes approches qui touchent aux statistiques, bien entendu, mais aussi à l'inférence statistique, à l'analyse de données complexes, à la théorie de l'information, à la mécanique statistique, et à l'algorithmique. L'apprentissage est un axe de recherche très vaste et qui intéresse plusieurs communautés de scientifiques. D'une part, des utilisateurs qui ont à leur disposition des jeux de données importants, font face à des problèmes réels de décision, de prédiction ou de précision, toutes disciplines confondues. Par exemple, on peut citer des applications dans le domaine de la prévision météorologique, en économie, en agronomie ou bien en médecine. Ces utilisateurs souhaitent des outils performants pour résoudre leurs problèmes spécifiques. D'autre part, les théoriciens sont capables de créer des modèles qui s'adaptent à un grand nombre de problèmes et à partir de n'importe quel jeu de donnés. Leur but est de construire les meilleurs modèles capables de prédire, décider ou de classer (sous certaines hypothèses), de faire leurs études théoriques, de les implémenter algorithmiquement et informatiquement, et de les améliorer.

Dans l'exemple introductif de ce chapitre, l'apprentissage est illustré par l'acquisition de la lecture chez les enfants. A partir d'exemples de lettres et de chiffres présents dans un livre de lecture et écrits avec une certaine police, l'enfant sera capable de reconnaître toute autre lettre, écrite dans la même police que celle de son livre ou non. En apprenant à lire, l'enfant construit alors une règle de correspondance entre un graphe (le symbole, la lettre écrite) et la signification de celle-ci (lettre 'A', 'B', 'C', etc...). Le livre de lecture est alors constitué d'exemples : une collection de graphes auxquels sont associées des significations, *i.e.* des étiquettes ou des labels.
En apprentissage mathématique, nous pouvons considérer les exemples comme étant des réalisations d'une suite de variables aléatoires $D_n = \{(X_i, Y_i), i = 1, \dots, n\}$ telles que :

- $(X_i, Y_i)$ pour $i = 1, \dots, n$ sont indépendantes et identiquement distribuées (i.i.d) de la loi $\mathscr{P}$ inconnue,
- $X_i \in \mathscr{X}$ pour $i = 1, \dots, n$ sont appelées **variables d'entrée** ou **instances**,
- $Y_i \in \mathscr{Y}$ pour $i = 1, \dots, n$ sont appelées **variables de sortie** ou **étiquettes** ou **classes**.

D'une manière générale, nous ne faisons aucune restriction sur l'espace $\mathscr{X}$. Dans le cadre de cette thèse, nous supposons que $\mathscr{X} \in \mathbb{R}^s$ avec $s \in \mathbb{N}^*$. Chaque entrée $x \in \mathscr{X}$ représente donc un vecteur de $\mathbb{R}^s$ où chaque composante est une caractéristique de $x$.

**L'apprentissage statistique :** l'apprentissage statistique consiste, à partir de l'échantillon d'apprentissage $D_n$, à prédire la variable de sortie $Y$ d'une nouvelle observation $X$ où $(X, Y)$ est de loi $\mathscr{P}$ indépendante de $D_n$ : nous cherchons à modéliser, à partir de l'échantillon d'observations, le lien qui existe entre la variable $X$ et la variable $Y$. Autrement dit, nous cherchons à construire une

fonction $f : \mathscr{X} \to \mathscr{Y}$ telle que, pour toute nouvelle observation $x \in \mathscr{X}$, sa prédiction $f(x)$ soit la plus proche possible de la valeur de $y$ que l'on aurait observée !

**Régression *vs* classification :** lorsque l'espace des réponses $\mathscr{Y}$ est continu, inclus dans $\mathbb{R}$, on parle alors de problème de **régression**. Dans le cas où l'espace des réponses est de cardinalité finie, on parle alors de **classification** [1]. Cette différenciation entre régression et classification est la plus courante. Néanmoins, d'autres différenciations sont possibles et même, il n'y a parfois aucune différenciation. Nous discutons de ce sujet au chapitre 3.
Dans le cadre de cette thèse, les problèmes de classification auxquels nous nous intéressons sont uniquement des problèmes de classification binaire, *i.e.* que la réponse $Y$ ne peut prendre que deux valeurs que nous notons $+1$ et $-1$, soit $Y = \{+1, -1\}$.
Dans le cas où l'ensemble $Y$ est de cardinalité supérieure à deux, alors le problème de classification peut se ramener à une collection de problèmes de classification binaire. Nous pouvons par exemple utiliser la méthode de l'apprentissage *un contre tous* : il suffit d'apprendre chaque classe contre toutes les autres, et en interrogeant tous les classifieurs binaires, l'affectation est alors faite à celui qui attribue l'exemple à sa classe majoritaire de la manière la plus forte.

**Espace d'hypothèses :** le problème de l'apprentissage statistique consiste donc à obtenir une fonction $\hat{f}$ dont le résultat soit le plus proche possible de ce qu'aurait fourni la loi de probabilité $\mathscr{P}$ inconnue sur tout l'espace $\mathscr{X}$. Pour obtenir cette fonction $\hat{f}$, nous allons la chercher dans un espace d'hypothèses fonctionnel $\mathbb{H}$ que l'on nomme couramment l'**espace d'hypothèses**.

**Fonctions de perte (ou fonction de coût) :** lorsque nous disposons d'une fonction $\hat{f} \in \mathbb{H}$ de l'espace fonctionnel, nous pouvons évaluer la qualité de celle-ci grâce à une **fonction de coût** (ou **fonction de perte**) :

$$\begin{aligned} l_{\hat{f}} : \mathscr{X} \times \mathscr{Y} &\to \mathbb{R}^+ \\ (x,y) &\to l\left(\hat{f}(x), y\right). \end{aligned}$$

Cela représente le coût de prédire $\hat{f}(x)$ au lieu de $y$. En régression, plus l'approximation de $y$ par $\hat{f}(x)$ est proche, plus le coût sera petit et inversement. Dans le cas de la classification, le coût est nul lorsque $y = \hat{f}(x)$. Dans le cas de la régression, les fonctions de coût classiques sont :

1. la fonction de perte charnière ( ou *hinge loss* en anglais) : $l_{\hat{f}}(x,y) = \max\left(1 - y\hat{f}(x), 0\right)$,
2. la fonction de perte quadratique, ou perte $L2$ : $l_{\hat{f}}(x,y) = (\hat{f}(x) - y)^2$,
3. la fonction de perte $L1$ : $l_{\hat{f}}(x,y) = \left|\hat{f}(x) - y\right|$,
4. la fonction de perte de Huber : $l_{\hat{f}}(x,y) = \begin{cases} \frac{1}{2\varepsilon}(\hat{f}(x) - y)^2 & \text{si } |\hat{f}(x) - y| \leq \varepsilon \\ 0 & \text{sinon} \end{cases}$
5. la fonction de perte de Vapnik : $l_{\hat{f}}(x,y) = \begin{cases} 0 & \text{si } |\hat{f}(x) - y| \leq \varepsilon \\ |\hat{f}(x) - y| - \varepsilon & \text{sinon} \end{cases}$

Dans le cas de la classification, la principale fonction de coût est la fonction indicatrice définie par : $l_{\hat{f}}(x,y) = \mathbb{1}\left(-y\hat{f}(x) \leq 0\right)$.

**Erreur en généralisation (ou coût en généralisation) :** la fonction de coût définie précédemment permet de mesurer localement la qualité d'une fonction $\hat{f}$ de l'espace d'hypothèses. Pour mesurer la qualité d'une hypothèse $\hat{f}$ sur l'ensemble de définition, on définit alors l'**erreur en généralisation** ou **coût en généralisation** sur l'ensemble par :

$$\mathscr{R}\left[\hat{f}\right] = \mathbb{E}\left[l_{\hat{f}}(X,Y)\right] = \int_{\mathscr{X}\times\mathscr{Y}} l_{\hat{f}}(x,y)\, d\mathscr{P}(x,y). \tag{1.1}$$

1. Le mot « classification » est utilisé ici dans sa terminologie anglaise, en français, le terme adéquat serait discrimination.

Lorsqu'en classification on utilise la fonction de perte indicatrice, alors l'erreur en généralisation correspond à la probabilité de mauvaise prédiction de la variable $Y$ par l'hypothèse $\hat{f}$sur tout l'espace, *i.e.* $\mathscr{R}\left[\hat{f}\right] = \mathbb{P}\left[\hat{f}(X) \neq Y\right]$.
Le choix de la meilleure hypothèse dans l'espace fonctionnel $\mathbb{H}$ revient donc à choisir celle qui minimise l'erreur en généralisation. L'erreur en généralisation utilise dans sa définition la densité de probabilité $\mathscr{P}$ qui est inconnue. Par conséquent, nous ne pouvons évaluer cette erreur. Une méthode naturelle pour l'estimer consiste à considérer le risque empirique sur les données d'apprentissage, autrement appelé aussi erreur d'apprentissage. Nous revenons plus en détails sur cette méthode dans le paragraphe 1.2.

**Erreur d'apprentissage :** nous pouvons estimer l'erreur réalisée par une fonction $\hat{f}$ sur l'ensemble d'apprentissage $D_n$ en estimant l'**erreur d'apprentissage** ou **erreur empirique** définie par :

$$\hat{\mathscr{R}}_n\left[\hat{f}\right] = \frac{1}{n}\sum_{i=1}^{n} l_{\hat{f}}\left(x_i, y_i\right). \tag{1.2}$$

## 1.2 Apprentissage statistique et classification

Cette section introduit les concepts propres aux problèmes de classification et présente les principaux résultats théoriques en classification statistique. Nous ne nous intéressons qu'au cas de la classification binaire, *i.e.* au cas où l'espace des variables de réponse $\mathscr{Y}$ est restreint à l'ensemble $\{0,1\}$.

### 1.2.1 Théorie de la minimisation du risque empirique

Cette section fournit des éléments pour choisir « correctement » une fonction dans l'espace fonctionnel.

**Règle de Bayes :** la règle de Bayes est l'application $f^*$ définie sur $\mathscr{X}$ par :

$$\begin{array}{rccll} f^*: & \mathscr{X} & \to & \mathscr{Y} & \text{où } \eta(x) = \mathbb{P}\left[Y = 1 | X = x\right]. \\ & x & \mapsto & \mathbb{1}\left(2\eta(x) - 1 > 0\right) & \end{array}$$

Cette fonction est la meilleure que l'on puisse espérer. Elle consiste à renvoyer la valeur 1 à $x$ lorsque $\mathbb{P}\left[Y+1|X=x\right] > \frac{1}{2}$ et 0 sinon. Cependant, cette fonction de Bayes dépend de la probabilité conditionnelle $\mathbb{P}\left[Y+1|X=x\right]$, et donc de la loi $\mathscr{P}$ inconnue. Par conséquent, cette règle est aussi inconnue !

**Principe de minimisation du risque empirique :** le problème de l'apprentissage statistique est d'estimer le meilleur classifieur de la collection $\mathbb{H}$ d'hypothèses appelé oracle et défini par :

$$f^*_{\mathbb{H}} = \arg\min_{f \in \mathbb{H}} \mathscr{R}\left[f\right].$$

Ce classifieur n'est pas mesurable car il dépend lui aussi de la loi $\mathscr{P}$ inconnue. L'idée naturelle proposée par Vapnik et Chervonenkis (1971) est alors de considérer le risque empirique.

Le classifieur ERM (Empirical Risk Minimizer) , s'il existe, noté $\hat{f}_n$, est celui qui minimise l'équation (1.2) sur la classe $\mathbb{H}$. Autrement dit :

$$\hat{f}_n \in \arg\min_{f \in \mathbb{H}} \hat{\mathscr{R}}_n\left[f\right].$$

La fonction $\hat{f}_n$ est une fonction optimale sur la base d'exemples $D_n$, cependant cela ne donne aucune garantie quant à son comportement optimal sur de nouveaux exemples. Ceci est le problème du sur-apprentissage : l'hypothèse est sur-adaptée à nos données. L'adaptation se révèle néfaste dans deux cas de figure :

- lorsque les données sont bruitées,
- lorsque le concept $f$ n'appartient pas à l'espace d'hypothèses $\mathbb{H}$ considéré.

Le cas limite est celui de l'apprentissage par cœur des données d'apprentissage pour lequel aucune garantie de bonne généralisation ne peut être établie.
Cependant, intuitivement, lorsque le nombre d'observations est grand, $\hat{f}_n$ s'approche de $f^*_{\mathbb{H}} = \inf_{g\in\mathbb{H}} \mathscr{R}[g]$. Pour une fonction $f$ fixée, $\hat{\mathscr{R}}_n[f] \to \mathscr{R}[f]$ d'après la loi des grands nombres. La consistance de cet estimateur revient donc à établir une loi des grands nombres fonctionnelle, c'est à dire une convergence uniforme sur l'espace $\mathbb{H}$.

Lorsque la fonction cible $f$ appartient à l'espace d'hypothèses $\mathbb{H}$, et que les données d'apprentissage $D_n$ ne sont pas bruitées, nous avons alors l'égalité suivante :

$$\mathscr{R}^* = \inf_{g\in\mathbb{H}} \mathscr{R}[g] = \mathscr{R}[f] = 0.$$

Supposons de plus que l'espace d'hypothèses $\mathbb{H}$ est de cardinalité finie. Alors, en notant $\hat{f} = \lim_{n\to+\infty} \hat{f}_n$, Li et Vitanyi (1993) ont démontré que :

$$\mathbb{P}\left[\mathscr{R}\left[\hat{f}\right] < \varepsilon\right] \leq \mathscr{C}ard(\mathbb{H})\, e^{-n\varepsilon}$$

et

$$\mathbb{E}\left[\mathscr{R}\left[\hat{f}_n\right]\right] \leq \frac{1+\log(\mathscr{C}ard(\mathbb{H}))}{n}.$$

Dans le cas où la fonction cible $f$ appartient à l'espace $\mathbb{H}$, alors $\mathscr{R}^* = 0$, et cela implique que l'erreur en généralisation est quasiment nulle lorsque la taille $n$ de l'échantillon est grande devant $log(\mathscr{C}ard(\mathbb{H}))$.
Dans le cas où l'hypothèse d'une erreur minimale nulle n'est pas réaliste, une théorie a été définie pour s'en affranchir : la théorie de Vapnik-Chervonenkis.

Les résultats précédents se basent sur l'estimation de la meilleure fonction de $\mathbb{H}$ par rapport à l'échantillon d'apprentissage $D_n$. Une estimation très satisfaisante en terme d'erreur d'apprentissage consiste à apprendre par cœur ces données d'apprentissage (phénomène de sur-apprentissage ou d'overfitting). L'hypothèse apprise sera alors bonne sur l'échantillon, mais répondra au hasard sur des nouvelles données, et aura donc une mauvaise capacité de généralisation. L'hypothèse retenue en utilisant l'échantillon $D_n$ ne serait pas nécessairement celle retenue avec un autre échantillon. Un algorithme d'apprentissage sera dit PAC, Probablement Approximativement Correct, s'il essaye de borner la différence de comportement entre le jeu de données d'apprentissage et de nouvelles données. La borne est une borne probabiliste qui dépend du nombre de points $n$ de l'échantillon et non de leur configuration spatiale.

**Apprentissage PAC :** une procédure d'apprentissage $\mathscr{A}$ pour $\mathscr{Y} = \{-1, 1\}$ est dite PAC (Probablement Approximativement Correct) si :

$$\forall \varepsilon > 0, \lim_{n\to+\infty} \sup_{P\in\mathscr{P}'} \mathbb{P}\left[\left|\hat{\mathscr{R}}_n - \mathscr{R}^*\right| > \varepsilon\right] = 0$$

où $\mathscr{P}'$ est l'ensemble des mesures de probabilités sur $\mathscr{X}\times\mathscr{Y}$.

L'apprentissage statistique classique réalise l'estimation d'une fonction cible à partir d'un échantillon d'observations de celle-ci. L'apprentissage PAC s'intéresse aux échantillons sur lesquels l'apprentissage est fait, et il donne des résultats de convergence de l'estimation quel que soit l'échantillon utilisé.

Autrement dit, quelle que soit la précision de l'estimation voulue, quelle que soit la confiance voulue en cette estimation, un algorithme d'apprentissage PAC pourra toujours atteindre cette estimation avec une suite finie croissante d'exemples.

### 1.2.2 Théorie de Vapnik-Chervonenkis

Soit $\mu$ une mesure de probabilité de $(X,Y)$ $\mathbb{R}^s \times \{-1,1\}$. Soit $\mu_n$ une mesure empirique basée sur $n$ réalisations de $\mu$, *i.e.* sur une base d'apprentissage $D_n$. Soit $A$ un ensemble mesurable de $\mathbb{R}^s \times \{-1,1\}$, alors $\mu(A) = \mathbb{P}[(X,Y) \in A]$ et $\mu_n(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}((X_i,Y_i) \in A)$. Soit $\mathscr{A}$ la classe de $\mathscr{X}$ de tous les ensembles de la forme :

$$\{\{x \in \mathscr{X} | g(x) = 1\} \times \{-1\}\} \cup \{\{x \in \mathscr{X} | g(x) = -1\} \times \{1\}\} \forall g \in \mathbb{H}$$

Vapnik (1995) a démontré que l'on a alors la relation suivante :

$$\sup_{g \in \mathbb{H}} \left| \hat{\mathscr{R}}_n[g] - \mathscr{R}[g] \right| = \sup_{A \in \mathscr{A}} |\mu_n(A) - \mu(A)|$$

**Coefficient de pulvérisation :** étant donné un espace fonctionnel $\mathbb{H}$, le $n^{\text{ième}}$ coefficient de pulvérisation de $\mathbb{H}$ est donné par :

$$\mathscr{S}(\mathbb{H},n) = \sup_{(x_1,\ldots,x_n) \in \mathscr{X}^n} \mathscr{C}ard((g(x_1),\ldots,g(x_n)) | g \in \mathbb{H})$$

Le coefficient de pulvérisation $\mathscr{S}[\mathbb{H},n]$ représente le nombre de manières dont on peut répartir les classes $g(x_1),\ldots,g(x_n)$ sachant que l'on dispose de $n$ exemples et que la fonction cible est dans $\mathbb{H}$. Plus ce nombre sera petit, plus il sera alors facile de choisir la « bonne » fonction adéquate de l'espace $\mathbb{H}$, et ce d'autant plus que la taille $n$ de l'échantillon est petite. Ce coefficient représente une mesure de la richesse de l'espace fonctionnel $\mathbb{H}$.
On peut également définir des coefficients de pulvérisation pour des fonctions à valeur dans $\mathbb{R}$ : il s'agira des coefficients de P-pulvérisation et de $\gamma$-pulvérisation.

La fonction cible étant à valeurs dans $\{-1,1\}$ , on a alors $\mathscr{S}(\mathbb{H},n) \leq 2^n$. En effet, si l'on est complètement libre de choisir les classes des $x_i$, il y a $2^n$ combinaisons possibles. Si $\mathscr{S}(\mathbb{H},n) = 2^n$, alors il existe un ensemble d'apprentissage de taille $n$ tel que celui-ci ne nous donne aucune information sur le choix de la fonction $f$ dans $\mathbb{H}$.

**Dimension de Vapnik-Chervonenkis :** soit $\mathbb{H}$ un espace fonctionnel de fonctions de $\mathscr{X}$ à valeurs dans $\{0,1\}$ . La dimension de Vapnik-Chervonenkis de l'espace $\mathbb{H}$, si elle existe, est définie par :

$$\mathscr{VC}_{\mathbb{H}} = \max\{n \in \mathbb{N}^* | \mathscr{S}(\mathbb{H},n) = 2^n\}.$$

Si $\forall n \in \mathbb{N}^*$, $\mathscr{S}(\mathbb{H},n) = 2^n$, alors $\mathscr{VC}_{\mathbb{H}} = +\infty$.

La dimension de Vapnik-Chervonenkis représente le nombre de points minimal que $\mathbb{H}$ ne peut pas pulvériser.
Si la dimension de Vapnik-Chervonenkis est infinie, alors quel que soit l'exemple qu'on rajoute, l'espace d'hypothèses sera tellement vaste qu'il sera possible de trouver une fonction compatible

avec l'échantillon d'apprentissage et avec tout nouvel exemple. Autrement dit, toute généralisation devient impossible.
A l'inverse, une dimension de Vapnik-Chervonenkis finie implique qu'avec un certain nombre d'exemples (égal à la $\mathscr{VC}$), il est possible d'identifier dans $\mathbb{H}$ une meilleure hypothèse.

**Lemme de Sauer :** si la VC-dimension $\mathscr{VC}$ est bornée par $V$, *i.e.* $\mathscr{VC}_{\mathbb{H}} < V < +\infty$, alors $\mathscr{S}(\mathbb{H},n) \leq \sum_{i=1}^{V} C_n^i < \left(\frac{en}{V}\right)^V, \forall n \geq V \geq 1.$
Ce lemme permet de borner les coefficients de pulvérisation avec la dimension de Vapnik-Chervonenkis.

**Inégalité de Vapnik-Chervonenkis :** pour toute mesure $\mu$ et pour tout ensemble $\mathscr{A}$ définis comme au début du chapitre (1.2.2), alors $\forall \varepsilon > 0$, nous avons la relation :

$$\mathbb{P}\left[\sup_{A\in\mathscr{A}} |\mu_n(A) - \mu(A)| > \varepsilon\right] \leq 8\mathscr{S}(\mathscr{A},n)\exp\left(-\frac{n\varepsilon^2}{32}\right).$$

La preuve de ce résultat se trouve dans Devroye et Rosenblatt (1982).

Grâce au lemme de Sauer énoncé précédemment, nous pouvons borner les coefficients de pulvérisation, et par conséquent la déviation de l'erreur en fonction de la dimension de Vapnik-Chervonenkis. Ce résultat montre que la minimisation du risque empirique permet à un algorithme de converger vers une hypothèse d'erreur optimale avec une vitesse exponentielle, et ceci même dans le pire des cas. Cependant, il est nécessaire de pouvoir calculer $\mathscr{S}(\mathscr{A},n)$, que l'on peut également borner (grâce au lemme de Sauer) par la dimension de Vapnik-Chervonenkis. Dans le cas où celle-ci est infinie, ce résultat ne nous fournit plus de garanties sur la qualité de la fonction apprise.

La dimension de Vapnik-Chervonenkis représente une mesure de la qualité de l'espace d'hypothèses $\mathbb{H}$. Si la fonction cible appartient à un espace fonctionnel de dimension $\mathscr{VC}$ de Vapnik-Chervonenkis finie, alors celle-ci est $\varepsilon-$ PAC apprenable avec un échantillon aléatoire d'apprentissage dont le nombre d'exemples est de l'ordre de $\mathscr{O}\left(\frac{\mathscr{VC}}{\varepsilon}\right), \forall \varepsilon > 0.$

La dimension de Vapnik-Chervonenkis n'est pas la seule mesure possible : les nombres de couverture de l'espace d'hypothèses sont également une autre mesure.

**Nombre de couverture :** géométriquement et implicitement, la taille d'un espace fonctionnel peut aussi être mesurée par une distance (fonctionnelle) maximum entre deux fonctions de cet espace. Soit $\rho$ une distance définie sur $\mathbb{H}$. Notons $\mathbf{B}(h*,\varepsilon) = \{h* \in \mathbb{H} | \rho(h,h*) \leq \varepsilon\}$ la boule fermée de centre $h^* \in \mathbb{H}$ et de rayon $\varepsilon$.
Une $\varepsilon$-couverture de $\mathbb{H}$ est un ensemble $\mathscr{H}$ fini de $\mathbb{H}$ tel que $\mathbb{H} \subseteq \bigcup_{h\in\mathscr{H}} \mathbf{B}(h,\varepsilon)$.
Le nombre $\varepsilon$ de couverture est le nombre minimal $\mathbf{N}(\varepsilon,\mathbb{H})$ tel qu'il existe une $\varepsilon$-couverture de cardinalité $\mathbf{N}(\varepsilon,\mathbb{H})$.
Sauer (1972) a démontré que les espaces fonctionnels à dimension de Vapnik-Chervonenkis finie possèdent des nombres de couverture « petits ». Autrement dit, si la dimension de Vapnik-Chervonenkis est égale à $\mathscr{VC}$, alors le nombre $\varepsilon$ de couverture est de l'ordre de $\mathscr{O}\left(\frac{1}{\varepsilon^{\mathscr{VC}}}\right)$. Une classe d'hypothèses telle que les nombres d'$\varepsilon$ couverture sont tous finis est une classe qui est PAC apprenable.

## 1.3 Apprentissage actif

A la section précédente, nous avons défini l'apprentissage statistique comme la recherche d'un lien fonctionnel entre deux variables $X$ et $Y$ en se basant sur un échantillon aléatoire d'exemples : nous avions alors émis l'hypothèse que les points d'apprentissage sont issus de variables aléatoires indépendantes et identiquement distribuées. Dans ce contexte, l'apprenant ou l'algorithme d'apprentissage se « contente » d'apprendre la meilleure règle possible en fonction des données dont il dispose.

De même qu'un étudiant interagit avec son professeur en lui posant des questions afin d'augmenter la vitesse et la qualité de son apprentissage, de même il est possible d'élargir ce contexte d'apprentissage en autorisant des interactions entre l'apprenant et un oracle expert pour enrichir les données d'apprentissage. Les exemples sur lesquels l'apprentissage est alors réalisé ne sont plus aléatoires et indépendants. Cette méthode d'apprentissage, utilisée dans de nombreuses situations réelles, s'appelle l'apprentissage actif.

En fonction des interactions entre l'apprenant et l'oracle, il est possible de distinguer différentes formes d'apprentissage actif. Dans cette section, nous présentons ces différentes formes et faisons le lien qui existe entre ces formes et le cadre d'apprentissage dans lequel nous nous situons. Nous commençons par présenter une caractérisation de l'apprentissage actif selon le type d'échantillonnage utilisé : apprentissage sélectif *vs* apprentissage génératif. Notons que dans cette section, nous ne définissons que ces cadres d'échantillonnage et d'apprentissage. Nous ne présentons aucune méthode. Un état de l'art des méthodes de sélection est réalisé au chapitre 6. Dans une deuxième partie, nous nous intéressons à la méthode séquentielle d'échantillonnage en classification active. Dans une troisième partie, nous rappelons des résultats sur l'apprentissage actif en classification avec du bruit. Dans une quatrième section, nous rappelons les résultats d'apprentissage actif en régression. Enfin, dans une cinquième et dernière partie, nous présentons le contexte particulier du « Query Learning ».

### 1.3.1 Échantillonnage sélectif *vs* échantillonnage adaptatif

L'apprentissage actif suppose que l'apprenant peut questionner un expert. Nous pouvons alors distinguer deux types d'apprentissage en fonction des questionnements possibles : soit l'apprenant peut poser n'importe quelle question, soit l'apprenant ne peut poser qu'une question parmi un ensemble de questions possibles. En apprentissage mathématique, les questions concernent l'étiquetage d'instances présentées à un oracle expert.

Castro *et al.* (2005) réalisent mathématiquement cette distinction en apprentissage actif : ils distinguent ainsi l'échantillonnage sélectif de l'échantillonnage adaptatif. La principale différence entre ces deux approches est la nature des exemples présentés à l'expert.

Dans le cas de l'échantillonnage sélectif, ou *selective sampling*, la stratégie d'apprentissage actif n'observe qu'une partie restreinte de l'espace des entrées, matérialisée par des instances. Pour illustrer cette approche, l'image d'un « sac » d'instances pour lesquelles la stratégie active peut demander à l'oracle expert les étiquettes associées est généralement employée.

Dans le cas de l'échantillonnage adaptatif, ou *adaptative* ou *blind sampling* en anglais, les instances à présenter à l'oracle ne sont pas restreintes et la stratégie d'apprentissage actif peut explorer tout l'espace des entrées, à la recherche de zones à échantillonner finement.
L'échantillonnage adaptatif peut être vu comme un passage à la limite de l'échantillonnage sélectif. Théoriquement, cela est possible : il suffit de générer une grille de pas régulier sur l'espace des

entrées. En faisant tendre le pas vers zéro, le nombre de points de la grille tend vers l'infini, et la grille couvre tout l'espace. Il ne reste plus qu'à sélectionner les points les plus « informatifs ». Néanmoins, en pratique, cette approche n'est pas réalisable : en effet, pour permettre de sélectionner ou non chacune des instances candidates, elle requiert de nombreux calculs, qui, le plus souvent, sont redondants. Le nombre de calculs nécessaire est alors, le plus souvent, hors de portée informatique dans des cas pratiques (en temps de calcul et en espace mémoire). En pratique, il est donc nécessaire de distinguer les méthodes d'échantillonnage sélectif des méthodes d'échantillonnage adaptatif.

Dans cette thèse, sauf mention du contraire, nous nous situons dans le cas de l'apprentissage de l'« adaptative sampling ».

### 1.3.2 Batch active learning en classification

En apprentissage statistique, les points d'apprentissage sont issus de variables aléatoires indépendantes et identiquement distribuées. Nous pouvons voir cet apprentissage comme un apprentissage séquentiel, dans le sens où nous disposons de points étiquetés, nous faisons un modèle d'apprentissage. Puis, nous obtenons un nouveau point étiqueté, et nous itérons l'algorithme d'apprentissage, etc ...

Nous avons précédemment défini l'apprentissage actif comme une forme d'apprentissage dans lequel l'apprenant, ou l'algorithme d'apprentissage, est capable d'interagir avec un oracle expert et de choisir ces exemples. Dans cette théorie de l'apprentissage actif, Hoi *et al.* (2006), puis Sugiyama et Rubens (2008), définissent la procédure du « batch active learning » dans lequel, à chaque itération, l'algorithme doit choisir $\lambda$ exemples parmi un ensemble d'instances candidates.
Le batch active learning est donc une généralisation de l'apprentissage actif séquentiel, *i.e.* le cas où $\lambda = 1$. Cette approche est particulièrement intéressante dans deux situations :

1. lorsque l'étiquetage d'instances est coûteux en temps de calcul et que le temps d'étiquetage d'instances n'est pas linéaire en fonction de leur nombre. Cela s'avère par exemple être le cas dans des constructions ou études de méta-modèles en physique, où l'étiquetage d'une instance est réalisé à travers des processus d'optimisations numériques qui nécessitent plusieurs heures ou journées de calcul. L'utilisation d'ordinateurs en parallèle permet alors d'accélérer l'étiquetage d'un ensemble d'instances. Dans un exemple plus concret, les scientifiques du LISC étudient la viabilité de systèmes complexes. Dans ce but, ils apprennent de manière itérative des surfaces de réponse de modèles. Chaque apprentissage utilise les résultats des apprentissages précédents. Néanmoins, à chaque apprentissage, il est nécessaire de générer de nouveaux points d'apprentissage. L'étiquetage de chaque point est indépendant des autres et est très coûteux en temps car nécessite beaucoup de simulations. Cependant, ces étiquetages peuvent être réalisés de manière séquentielle, diminuant ainsi significativement la durée de fonctionnement d'un algorithme de viabilité.
2. l'apprentissage du modèle de classification nécessite un nombre d'optimisation ou de calculs élevé, le rendant couteux en temps de calcul,alors que le temps d'étiquetage d'un point est négligeable.

Dans le cas où le temps de modélisation du problème de classification est très important par rapport à l'étiquetage, il est alors préférable de faire du batch active learning, en limitant le nombre de sollicitations de l'algorithme d'apprentissage. Rolet et Teytaud (2010) ont étudié l'apport de celui-ci et nous présentons dans la suite de cette section les résultats principaux de ces travaux.

Rappelons que l'objectif est de limiter le nombre d'appels à l'algorithme d'apprentissage. Pour comparer la stratégie du « batch active learning » à celle de l'apprentissage séquentiel, il est alors nécessaire de comparer, pour des échantillons d'apprentissage de taille identique, le nombre d'appels aux algorithmes d'apprentissage. Dans le cas séquentiel pour $\lambda$ points, nous appelons $\lambda$ fois

l'algorithme d'apprentissage. Le coût total d'apprentissage, $\mathscr{I}ter_1$, est alors linéaire en fonction du nombre de points $\lambda$. Sous l'hypothèse que l'algorithme d'apprentissage peut être étendu au batch active learning avec des lots d'exemples étiquetés de taille $\lambda$, le nombre d'itérations de cet algorithme est noté alors $\mathscr{I}ter_\lambda$. Le gain du batch active learning se mesure alors comme étant le rapport $\frac{\lambda \mathscr{I}ter_1}{\mathscr{I}ter_\lambda}$.

#### 1.3.2.1 Étude première

Kulkarni et Haussler (1993) ont démontré que, dans le cas de l'apprentissage actif séquentiel, l'hypothèse de dimension de Vapnik-Chervonenkis finie de l 'espace d'hypothèses $\mathbb{H}$ permettait d'obtenir des résultats plus intéressants que dans le cas de la dimension infinie. Nous supposerons donc dans la suite de cette section que cette dimension, égale à $\mathscr{VC}$, est finie. Balcan *et al.* (2008) montrent que, dans la majorité des cas, pour obtenir une estimation $\hat{f}$ de la fonction cible, avec un risque en généralisation $\mathscr{R}[f] < \varepsilon$ et une confiance supérieure à $1-\delta$, le nombre d'itérations de l'algorithme ( *i.e.* le nombre d'exemples) est de l'ordre de $\mathscr{O}\left(\mathscr{VC}\log\left(\frac{1}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$.

En batch active learning, pour des itérations à $\lambda$ exemples choisis, et un algorithme d'apprentissage $\mathscr{A}_\lambda$ délivrant une estimation $\mathscr{A}_\lambda[\mathbb{H}, D_n]$ de la fonction cible dans l'espace fonctionnel $\mathbb{H}$ à partir d'un échantillon $D_n$ de taille $n$ (multiple de $\lambda$), le nombre minimum d'itérations pour obtenir une précision $\varepsilon$ est égal à :

$$\mathscr{I}ter_\lambda(\varepsilon, \mathbb{H}, \mathscr{A}_\lambda) = \min\left\{n \in \mathbb{N} | \max_{f\in\mathbb{H}} \mathscr{R}[\mathscr{A}_\lambda[\mathbb{H}, D_n]] < \varepsilon\right\}.$$

Avec le formalisme PAC, et en prenant en compte l'ensemble des algorithmes d'apprentissage, le nombre minimal d'itérations nécessaires devient alors :

$$\mathscr{I}ter^*_\lambda(\varepsilon, \mathbb{H}) = \inf_{\mathscr{A}_\lambda} \mathscr{I}ter_\lambda(\varepsilon, \mathbb{H}, \mathscr{A}_\lambda).$$

#### 1.3.2.2 Borne inférieure du nombre d'appels d'apprentissage

Dasgupta (2006) démontre que dans le cas d'un espace d'hypothèses $\mathbb{H}$ à dimension $\mathscr{VC}$ de Vapnik-Chervonenkis finie, alors il existe des constantes $C, M > 0$ telles que, pour tout $\varepsilon > 0$, le nombre d'$\varepsilon$-couverture $\mathbf{N}(\varepsilon, \mathbb{H})$ est minoré par :

$$\forall \varepsilon > 0, \mathbf{N}(\varepsilon, \mathbb{H}) \geq \left(\frac{M}{\varepsilon}\right)(C * \mathscr{VC}).$$

Vidyasagar (1997) démontre que, en apprentissage séquentiel ( *i.e.* $\lambda = 1$), pour une précision $\varepsilon$, le nombre d'exemples est borné inférieurement par :

$$\mathscr{I}ter^*_1(\varepsilon, \mathbb{H}) \geq \left\lceil C.\mathscr{VC}.\log\left(\frac{M}{\varepsilon}\right)\right\rceil.$$

Rolet et Teytaud (2010) démontrent le théorème suivant dans le cas d'un espace d'hypothèses $\mathbb{H}$ à dimension $\mathscr{VC}$ de Vapnik-Chervonenkis finie (le résultat du paragraphe précédent s'applique alors). Ce résultat est asymptotique.

**Théorème.** *Borne inférieure en batch learning*
*Pour $\lambda > 1$, le nombre de points nécessaire pour un apprentissage de précision $\varepsilon$ est borné inférieurement par :*

$$\mathscr{I}ter^*_\lambda(\varepsilon, \mathbb{H}) \geq C.\mathscr{VC}.\log\left(\frac{M}{\varepsilon}\right)\frac{1}{\log(K)}$$

*où* $K = \begin{cases} K = \lambda^{\mathscr{VC}} & si\ \mathscr{VC} \geq 3 \\ K = \lambda^{\mathscr{VC}} + 1 & si\ \mathscr{VC} \leq 2 \end{cases}$ *. Ce nombre K représente une borne supérieure du nombre de façons de classifier* $\lambda$ *points dans un espace fonctionnel de dimension* $\mathscr{VC}$ *de Vapnik-Chervonenkis.*

Autrement dit,

$$\mathscr{I}ter^*_\lambda(\varepsilon, \mathbb{H}) \geq C.\mathscr{VC}.\log\left(\frac{M}{\varepsilon}\right)\frac{1}{\log(\lambda)}. \tag{1.3}$$

Ce théorème montre une diminution notable du nombre d'instances requises, et ce d'autant plus que la dimension de Vapnik-Chervonenkis de l'espace d'hypothèses est élevée (mais finie). L'équation 1.3 nous indique alors que la vitesse optimale est en $\mathscr{O}(\log(\lambda))$. Ce théorème démontre ainsi l'intérêt du batch active learning en pratique.

#### 1.3.2.3 Borne supérieure du nombre d'appels d'apprentissage

Rolet et Teytaud (2010) démontrent le théorème suivant dans le cas d'un espace d'hypothèses $\mathbb{H}$ à dimension $\mathscr{VC}$ de Vapnik-Chervonenkis finie (le résultat du paragraphe précédent s'applique alors).

**Théorème.** *Amélioration logarithmique en batch learning*
*Soient* $K = \lambda^{\mathscr{VC}} + 1, b \geq 1$ *et* $\lambda' = \lambda\frac{K^b - 1}{K - 1}$.
*Alors*

$$\mathscr{I}ter^*_{\lambda'}(\varepsilon, \mathbb{H}) \leq \left\lceil \frac{\mathscr{I}ter^*_\lambda(\varepsilon, \mathbb{H})}{b} \right\rceil. \tag{1.4}$$

Pour $b$ et $\lambda$ fixés, alors l'équation 1.4 implique que $\mathscr{I}ter^*_{\lambda'}(\varepsilon, \mathbb{H}) = \mathscr{O}\left(\frac{\mathscr{I}ter^*_\lambda(\varepsilon, \mathbb{H})}{\log(\lambda')}\right)$. Ce résultat stipule que l'amélioration de la vitesse est alors, dans tous les cas, au moins logarithmique.

**Théorème.** *Amélioration linéaire en batch active learning pour* $\lambda = \mathscr{VC}$
*Soit* $D > 0$*, en notant* $\mathbb{H}_D = \left\{[0,x]\,; x \in [0,1]^D\right\}$*. Nous avons alors* $\mathscr{VC}(\mathbb{H}_D) = D$*. Alors, pour* $M, M' > 0$ *indépendants de D :*

1. $\exists C > 0, \exists \varepsilon_0, \forall \varepsilon < \varepsilon_0, \mathscr{I}ter^*_1(\varepsilon, \mathbb{H}_D) \geq C.D\log\left(\frac{M}{\varepsilon}\right)$ ;
2. $\exists C' > 0, \exists \varepsilon_0, \forall \varepsilon < \varepsilon_0, \mathscr{I}ter^*_D(\varepsilon, \mathbb{H}_D) \leq C'.D\log\left(\frac{M'}{\varepsilon}\right)$.

Ce théorème, dû également à Rolet et Teytaud (2010), stipule donc que pour $\lambda = \mathscr{VC}$ la vitesse est linéaire pour la famille de fonctions $\mathbb{H}_D$ définie précédemment.

**Conclusion :** l'ensemble de ces travaux démontre que les apports d'un apprentissage en mode batch active learning en terme de réduction du nombre d'itérations de l'algorithme d'apprentissage sont alors :
- logarithmiques dans tous les cas ;
- linéaires lorsque $\lambda = \mathscr{VC}$ pour certaines familles de fonctions.

#### 1.3.2.4 Positionnement de nos travaux par rapport au « Batch Active Learning »

La section précédente présentait les résultats théoriques connus en apprentissage par lots, montrant, dans certains cas, l'apport de l'apprentissage batch par rapport à l'apprentissage classique.

Notre problématique de recherche s'intéresse à générer des points d'apprentissage de manière optimale afin d'augmenter la qualité d'apprentissage. Supposons que nous disposons d'un budget

pour étiqueter $n$ instances. Nous pouvons, par exemple, commencer par étiqueter $n_1$ instances, réaliser un premier apprentissage et, par la suite, exploiter cet apprentissage pour choisir les $n_2 = n - n_1$ instances restantes afin d'augmenter la qualité de ce dernier. Le choix de ces $n_1$ et $n_2$ points peut être vu alors comme étant des réalisations d'un apprentissage en mode batch. Notre thématique de recherche trouve donc sa place dans cette forme d'apprentissage d' « Active Learning ».

Notre problématique consiste à réaliser une génération de points, non pas séquentielle, mais qui n'est pour autant ni aléatoire, ni une méthode de sélection (par opposition à la génération). Avec ce point de vue, nos travaux peuvent donc être considérés comme étant à la frontière de cette forme d'apprentissage. Nos travaux sont plus algorithmiques et pratiques, et ne sont nullement une comparaison théorique entre l'apprentissage séquentiel et l'apprentissage en mode batch.

### 1.3.3 Apprendre en classification avec un oracle bruité

Dans un autre registre, Angluin et Laird (1988) et Kearns *et al.* (1994) (se basant sur Kearns (1993)) définissent le contexte de « Statistical Query Learning ». Cette dénomination est apparue dans les travaux de Feldman (2008) et de Feldman (2009). Bien que contenant le terme de « Query Learning », ce contexte est différent de celui présenté à la section 1.3.5.
Le contexte de « Statistical Query » est une approche spécifique de l'apprentissage PAC. Comme dans celui-ci, les données d'apprentissage sont des réalisations aléatoires, indépendantes et identiquement distribuées. Celles-ci sont alors présentées à un oracle qui fournit un label. Cependant, la réponse du label est soumise à un bruit, et l'oracle peut donc délivrer un label faux avec une probabilité $\eta$. Pour le lecteur intéressé, les travaux de Szörényi (2009) sont une bonne introduction à ce formalisme. Ces travaux s'attachent à donner un cadre théorique à ce type d'apprentissage. La théorie de Vapnik-Chervonenkis présentée au début de ce chapitre peut être aussi adaptée en fonction de données bruitées.

D'un point de vue algorithmique, Balcan *et al.* (2006) proposent l'algorithme d'Agnostic active learning, *alias* $A^2$, pour sélectionner les instances à présenter à l'oracle lorsque les données sont munies de bruit ou lorsque la fonction cible n'appartient pas à l'espace d'hypothèses. Hanneke (2007) étudie théoriquement cet algorithme et définit ses bornes de convergence. En utilisant ces travaux, ces deux auteurs s'associèrent dans Balcan *et al.* (2008) pour améliorer l'algorithme du $A^2$.

Kaariainen (2006) stipule néanmoins que lorsqu'on réalise de l'apprentissage sur des données, l'espace fonctionnel utilisé ne comporte pas en général la fonction cible. Par conséquent, chaque apprentissage doit être réalisé en présence de bruit et il étudie alors l'apprentissage avec un taux $\eta$ de bruit. Cependant, Castro et Nowak (2007) stipulent que lorsque $\eta > \frac{1}{2}$, les exemples d'apprentissages sont étiquetés très aléatoirement, et il n'est alors pas raisonnable de réaliser un apprentissage sur ces données. Dans le cas inverse, pour une valeur $x_i$ présentée, l'oracle fournit alors l'espérance de l'étiquette, *i.e.* $\mathbb{E}[y_i|x_i]$. Ces mêmes auteurs étudient l'effet du taux lorsque les instances sont proches des frontières de classification.

Dans notre problématique de recherche, nous supposons que les données sont non bruitées. Par conséquent, nous ne nous intéressons pas à ce type d'apprentissage dans la suite de cette thèse.

### 1.3.4 Apprentissage actif de fonctions ou régression

L'apprentissage actif n'est pas un cadre d'apprentissage propre à la classification : l'apprentissage actif de fonctions, ou régression, est un problème qui peut se rencontrer également. L'apprentissage actif de manière aveugle [2] en régression a été initié par Fisher (1951) alors même que le terme d'apprentissage n'était pas encore utilisé. Ce scientifique a développé la théorie des plans d'expériences factoriels, qui est encore très utilisée dans le milieu industriel pour étudier des codes de calculs. Dans un autre registre, Niederreiter (1988) a développé de nouveaux plans d'expérience en utilisant les suites à faible discrépance. L'utilisation de ces nouveaux plans d'expérience en apprentissage de fonctions a été initiée par Cervellera et Muselli (2003a). Dans le domaine industriel, ils ont été utilisés au CEA [3] par Feuillard (2007). Teytaud *et al.* (2007) les utilisent également pour réaliser de la programmation dynamique stochastique. Nous reviendrons sur cette méthodologie au cours du chapitre 2.

Une autre approche plus statistique a été proposée par Cohn *et al.* (1996). Celle-ci se base sur la minimisation de la variance de l'« appreneur » : en utilisant comme espace d'hypothèses des combinaisons de fonctions gaussiennes, ils peuvent estimer la variance attendue du modèle en étiquetant un point, et choisissent alors d'étiqueter celui qui maximise la variance.

Castro *et al.* (2005) étudient également l'apprentissage actif de fonctions dans deux espaces d'hypothèses particuliers : l'espace des fonctions *Hölder-smooth* et l'espace des fonctions *piecewise-constant*. En ajoutant en plus un bruit gaussien indépendant des étiquettes sur celles-ci dans un espace de Hölder, ils obtiennent des résultats théoriques de vitesse de convergence de type *minimax*. Néanmoins, ces résultats sont relativement décevants car l'hypothèse d'avoir des fonctions *Hölder-smooth* est une contrainte assez forte.
En dérivant ces *Hölder-smooth*, ils obtiennent des fonctions *piecewise-constant* qui correspondent à une classe particulière de fonctions de classification.

### 1.3.5 Un autre formalisme d'apprentissage : le « Query Learning »

Dans les sections précédentes, nous avons présenté formellement l'apprentissage statistique et nous avons alors défini l'apprentissage actif. Lors de conférences ou de soumissions à d'articles de revues scientifiques, certains évaluateurs ont eu un regard très critique sur nos travaux sur l'apprentissage actif et ont considéré que ceux-ci étaient un cas particulier du « Query Learning ». Selon nous, ce formalisme d'apprentissage est très différent.

Angluin (1988) introduit et formalise un nouveau concept d'apprentissage : le « Queries and Concept Learning », qui peut se traduire en Français par « Questionnement et Apprentissage de concept *(ou règles)* ». La problématique de cette approche est d'identifier un concept inconnu $L^*$ parmi un ensemble de concepts connus en interrogeant un oracle expert afin d'obtenir des informations sur le concept inconnu.

Cet apprentissage s'intéresse naturellement aux données numériques comme l'apprentissage statistique classique présenté précédemment. Cependant, il est particulièrement adapté et utilisé pour l'apprentissage symbolique ou apprentissage de concept. Nous entendons par apprentissage de concept un apprentissage de règles symboliques ou binaires de la vie courante qui se représente aisément par un arbre de décision. Les nœuds de l'arbre sont alors des questions et les feuilles correspondent à un état du concept. A titre d'illustration, nous pouvons chercher à savoir quelle est la règle permettant de savoir si : « je peux aller courir en haut du Puy de Dôme ou non » :

2. Le blind active learning.
3. Commissariat à l'Énergie Atomique et aux Énergies Altenatives.

– Est-ce que je dois dispenser un cours à des étudiants ou faire de la recherche ? Si oui, je ne peux pas aller courir !
– Sinon, fait-il nuit ? Si oui, je ne peux pas aller courir !
– Sinon, y-a-t-il beaucoup de neige ? Si oui, je ne peux pas aller courir !
– Sinon, est-ce que je suis habillé en jean et en chemise ? Si oui, je ne peux pas aller courir !
– ...
– Sinon, je peux aller grimper le Géant des Dômes !

Formellement, l'identification du concept inconnu $L^*$ se fait parmi un ensemble fini ou dénombrable de concepts $L_1, L_2, \ldots$ et chaque concept est défini sur un domaine qui lui est propre. Tous ces domaines sont néanmoins inclus dans un même ensemble $U$. Dans la suite de cette section, nous noterons de la même façon $L^i$ le i$^e$ concept que l'ensemble sur lequel il est défini. A partir d'un échantillon de données, nous sélectionnons un concept $L$ parmi les concepts possibles, et nous interrogeons ensuite l'oracle qui peut délivrer différentes réponses selon le type de questionnement utilisé. Les deux principales formes d'apprentissage sont alors :

– **Membership Query :** si $L$ est de même classe que le concept inconnu $L^*$, l'oracle délivre la réponse « oui », et « non » sinon ;
– **Equivalence Query :** si $L = L^*$, alors l'oracle délivre la réponse « oui », sinon il délivre un contre-exemple.

A ces deux formes viennent s'ajouter des autres formes plus exotiques qui sont :

– **Subset Query :** si $L \subseteq L^*$, alors l'oracle délivre la réponse « oui », sinon il délivre un contre-exemple de $L - L^*$ ;
– **Superset Query :** si $L \supseteq L^*$, alors l'oracle délivre la réponse « oui », sinon il délivre un contre-exemple de $L^* - L$ ;
– **Disjointness Query :** si $L \cap L^* = \emptyset$, alors l'oracle délivre la réponse « oui », sinon il délivre un contre-exemple de $L^* \cap L$ ;
– **Superset Query :** si $L \cup L^* = U \neq \emptyset$, alors l'oracle délivre la réponse « oui », sinon il délivre un contre-exemple de $L^* \cup L$.

Beaucoup d'algorithmes polynomiaux ont été créés pour identifier différents concepts : on pourra se référer par exemple à Angluin (2001) ou à Gavaldà (1993) pour des algorithmes sur le « Membership Query » et sur le « Equivalence Query ». Une classe d'algorithmes particulièrement adaptée à cet apprentissage par query est l'ensemble des arbres de décision binaire.

En apprentissage statistique classique, avec le formalisme PAC, la problématique usuelle est de connaître combien d'exemples sont nécessaires pour approcher une fonction cible. Des bornes d'approximation de ce nombre peuvent alors êtres calculées grâce à une définition de la complexité de la fonction cible : la dimension de Vapnik-Chervonenkis. On peut faire un parallèle en apprentissage de concepts, où il est également possible de définir un des estimateurs de la complexité de la règle. Cependant, cette définition n'est pas aussi évidente . Il existe autant de définitions que de formes de « Query », voire d'hypothèses d'apprentissage. Pour s'en convaincre, il suffit de lire les travaux pionniers de Balcázar *et al.* (2001), ou alors ceux récapitulatifs de Balcázar *et al.* (2007).

Ce formalisme d'apprentissage diffère de celui dans lequel nous nous sommes placés, à savoir l'apprentissage actif génératif. Certaines écoles scientifiques ne distinguent pas les subtilités qu'il y a entre ces deux apprentissages. Les deux différences importantes résident dans les faits que :

– dans notre apprentissage, l'oracle ne délivre jamais de contre-exemples ;
– en Query Learning, nous ne pouvons jamais fournir une instance à l'oracle pour qu'il nous l'étiquette.

Enfin, un argument plus anedoctique est de regarder la littérature de futurs chapitres de cette thèse : aucun article ne fait référence à un papier issu de ce contexte de « Query Learning ». Par conséquent, nous ne nous situons pas dans ce contexte et nous ne nous intéressons pas à celui-ci.

## 1.4 Conclusion : notre contexte de l'apprentissage actif

Dans cette thèse, afin d'approcher des variétés, nous nous plaçons dans le contexte d'apprentissage actif aveugle sans bruit. Autrement dit, nous nous situons dans le contexte suivant :

1. Initialement, nous ne disposons d'aucun échantillon d'apprentissage.
2. Nous supposons que le domaine $\mathscr{X}$ des instances $X$ d'apprentissage est restreint au carré unité en dimension $s$, *i.e.* $\mathscr{X} = I^s = [0,1]^s$.
3. Nous disposons d'un oracle-expert capable de délivrer l'étiquette de tout point que nous lui présentons.
4. Nous réalisons un apprentissage global, *i.e.* que nous ne faisons pas une juxtaposition d'apprentissages locaux.
5. L'oracle est capable de d'étiqueter instantanément plusieurs instances, autrement dit, nous pouvons faire de l'apprentissage en mode « batch ».
6. L'oracle délivre des étiquettes sans bruit.
7. Il existe une notion de budget d'étiquetage : nous ne pouvons présenter qu'un nombre fini prédéfini d'instances à l'oracle.

Dans ce contexte, il n'existe initialement pas de points d'apprentissage. La première étape pour apprendre consiste donc à générer les premiers points pour les présenter à l'oracle afin qu'il les étiquette, puis qu'on puisse apprendre. La première partie de ce manuscrit traite de la génération de ces premièrss points d'apprentissage.

# PREMIÈRE PARTIE

## INITIALISATION DES POINTS D'APPRENTISSAGE DE VARIÉTÉS

# - CHAPITRE 2 -

# INITIALISATION DANS LE CAS DE LA RÉGRESSION : LE CRITÈRE DE LA DISCRÉPANCE

## Sommaire



LA PREMIÈRE ÉTAPE pour étudier une variété ou pour construite un modèle en apprentissage actif consiste à générer les premiers points d'apprentissage. Dans ce chapitre, nous nous intéressons principalement à cette action d'échantillonnage (ou *experimental design* en anglais) dans le cas d'études de modèles de régression ou d'apprentissage de nappes. Cette étape détermine alors le premier ensemble d'apprentissage sur lequel est effectuée ensuite la construction itérative du modèle. Le nombre de point de l'espace générés à présenter à l'oracle pour être étiquetés, doivent être à la fois assez élevé en nombre afin d'obtenir une première approximation de la fonction cible de bonne qualité, mais il doit être aussi limitée lorsque le calcul de l'approximation est coûteuse

en temps ou en ressource calculatoire, ou lorsque la présentation de points à l'oracle est également limitée par un certain budget.

Idéalement, les zones de l'espace où la fonction varie beaucoup doivent bénéficier d'un échantillonnage plus important, alors que les zones où la fonction varie peu ne doivent bénéficier que d'un échantillonnage plus superficiel. Par exemple, la partie gauche de la fonction présentée à la figure 2.1 varie rapidement alors que la partie droite est beaucoup plus plate : il est alors préférable d'avoir beaucoup de points d'échantillonnage dans la partie gauche de l'espace des paramètres et peu de points dans la partie droite.

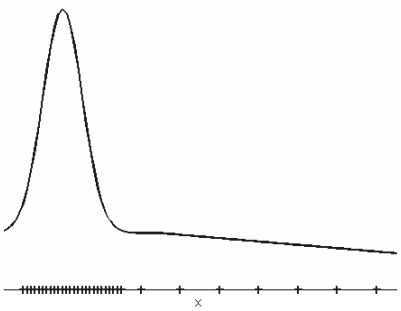


**FIGURE 2.1:** *Exemple d'échantillonnage adapté pour l'apprentissage d'une fonction réelle en dimension 1 : les points sont plus nombreux dans les zones où la fonction varie beaucoup.*

Lors de ce premier échantillonnage des points d'apprentissage, l'allure générale de la fonction cible est cependant inconnue. Afin d'obtenir en moyenne la meilleure approximation, les points doivent alors être générés « uniformément ». Classiquement, cette génération uniforme de points est réalisée en utilisant, soit une suite aléatoire uniforme, soit une grille régulière. Les résultats théoriques concernant l'utilisation des suites aléatoires uniformes ont été présentés au chapitre 1.

Dans ce chapitre, nous nous intéressons aux suites à discrépance faible comme méthode de génération « uniforme » de points. Pour cela, nous commençons par nous intéresser aux notions d'« uniformité » d'une suite et de discrépance d'une suite. Ensuite, nous présentons différentes suites à discrépance faible. Puis, nous nous intéressons au comportement de la discrépance avec la dimension de l'espace. Enfin, nous présentons l'utilisation de ces suites en apprentissage.
Notons que les méthodes utilisant de l'échantillonnage itératif ne sont pas étudiées dans ce chapitre et le seront dans la partie II de la thèse.

## 2.1 Uniformité d'une suite, discrépance et suite à discrépance faible

Dans les années 1940, en physique nucléaire, les physiciens travaillant sur l'arme nucléaire sont souvent confrontés aux calculs d'intégrales multiples compliquées. Comme leurs résolutions formelles sont impossibles, les physiciens estiment ces intégrales en utilisant des tirages aléatoires uniformes et la méthode de Monte-Carlo : ils tirent une suite de $n$ points de manière aléatoire indépendante et uniformément distribuée selon la distribution uniforme sur le domaine $D$ d'étude. Puis ils évaluent l'intégrale $\int_D f(x)\,dx$ par la moyenne des réalisations de $f$ sur cette suite, *i.e.* $\sum_{i=1}^{n} f(x_i)$.
L'évaluation de l'erreur commise dans l'approximation de l'intégrale peut être réalisée en utilisant

les théorèmes d'écart à la moyenne d'une variable aléatoire, *e.g.* le théorème de la limite centrale. Même s'il existe des méthodes de réduction de la variance de l'erreur, le résultat obtenu à partir de ces simulations reste stochastique. L'annexe C présente plus en détails cette méthode d'estimation d'intégrales.
Afin d'obtenir un résultat plus précis et déterministe, des chercheurs s'intéressent à la possibilité de remplacer les échantillons uniformes aléatoires impliqués dans les calculs par des suites déterministes ayant une « bonne » répartition : la méthode devient la méthode de Quasi Monte-Carlo et ces suites sont les suites à discrépance faible. Ces méthodes ont ensuite été reprises dans le domaine des mathématiques financières, toujours dans le but d'estimer des intégrales.

Dans cette section, nous nous intéressons à la notion d'« uniformité » d'une suite et définissons alors mathématiquement la notion de discrépance d'une suite. Dans un deuxième temps, nous nous intéressons à l'estimation de cette discrépance. Enfin, dans une troisième section, nous définissons des suites particulières : les suites à discrépance faible. Notons que la génération des suites à faible discrépance ne fait pas partie de cette section, mais sera expliquée en détails à la section 2.2.

### 2.1.1 Notions de discrépances d'un suite

Venant du latin *discrepantia*, qui signifie disconvenance ou divergence, la discrépance est un terme utilisé pour l'étude de désaccords. Dans un contexte mathématique, la discrépance d'une suite est une mesure de l'écart existant entre la suite et une situation de référence (généralement l'uniformité parfaite). Cette notion de discrépance est facilement compréhensible intuitivement, cependant il est difficile de la formaliser mathématiquement. Dans cette partie, nous nous intéressons à la notion d'uniformité d'une suite, définissons la discrépance mathématique de différentes manières et nous montrons que ces définitions sont équivalentes.

Dans ce chapitre, et dans les suivants, nous notons $I^s$ l'hyper-cube unité en dimension $s$, *i.e.* $I^s = [0,1]^s$.

**Suites uniformément équi-réparties :** considérons le problème de construire dans l'intervalle $[0;1]$ une suite de $n$ points $x_{(n)} = \{x_1,\dots,x_n\}$ qui soit uniforme, *i.e.* équirépartie. La manière la plus simple est de constituer une suite de $n$ points équidistants de la manière suivante : $x_i = \dfrac{i-1}{n-1}$ pour $i = 1,\dots,n$. Une représentation de cette configuration avec 5 points est présentée à la figure 2.2.

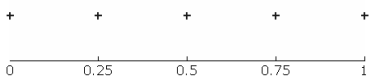



**FIGURE 2.2:** *Échantillonnage équi-réparti de* 5 *points en dimension* 1.

Considérons le même problème transposé au cas du carré unité de dimension 2 : la situation est nettement moins tranchée. En effet, il est alors possible d'énoncer plusieurs définitions de l'uniformité.

Choisissons par exemple le critère suivant : *une suite de n points* $x = \{x_1,\dots,x_n\}$ *est dite optimalement distribuée si elle minimise le* supremum *sur tous les rectangles* $P \in I^2$ *de la déviation* $|\#(P,x) - n\lambda(P)|$ *où* $\#$ *et la fonction de cardinalité (voir la section « Notations », page xv) et* $\lambda$ *est la mesure de Lebesgue, i.e. une mesure de volume*. Comme on s'attend à ce que la proportion des

points situés dans un rectangle P de $I^2$ soit proche de son aire, il semble en effet naturel de juger peu uniforme une séquence pour laquelle il est possible d'exhiber un rectangle avec une importante déviation. Cette mesure est non bornée et cette définition se comprend intuitivement.
Autrement dit, une suite $x = \{x_1, \ldots, x_n\}$ est uniformément équi-répartie dans $[0,1]^s$ si pour tout $a = (a_1, \ldots, a_s)$ de $[0,1]^s$ : $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{x_i \in [0,a_i]\}} = \lambda(a)$. Ce volume $\lambda(a)$ doit être calculé sur tous les rectangles de l'espace contenant l'origine.

Nous pouvons élargir la définition précédente à tous les rectangles de l'hypercube ne contenant pas nécessairement l'origine, et nous pouvons encore l'élargir à tous les convexes de l'hypercube. Nous obtenons ainsi différentes notions d'uniforme équi-répartition, et celles-ci ne sont alors pas bornées. A partir de celles-ci, nous pouvons définir de nouvelles définitions de la répartition uniforme : les discrépances. Dans la suite, nous définissons ces discrépances en commençant par la discrépance la plus « large » : la discrépance isotrope.

**La discrépance isotrope :** définissons $\mathbb{C}^s$ comme l'ensemble des convexes de $I^s$.
Notons formellement $\lambda$ la mesure de Lebesgue (associant la surface ou le volume pour les ensembles de dimension 2 ou 3) et # l'opérateur qui, pour une séquence $x = x_{(n)} = \{x_1, \ldots, x_n\}$ à $n$ éléments et un ensemble $P$, donne le nombre d'éléments de $x$ dans l'ensemble $P$.
La discrépance isotrope est définie par :

$$J_n(x) = J\left(x_{(n)}\right) = \sup_{P \in \mathbb{C}^s} \left| \frac{\#(P, x_{(n)})}{n} - \lambda(P) \right|. \tag{2.1}$$

Autrement dit, la discrépance isotrope d'une suite correspond à la mesure de l'écart entre la proportion de points présents dans tout convexe et la proportion du volume du convexe par rapport au volume total du domaine considéré. Un schéma représentatif de cette discrépance est présenté à la figure 2.3. Lorsque les points sont « bien » répartis [1], cet écart est petit et la discrépance est petite. A l'inverse, lorsque les points ne sont pas « bien » répartis, il existe alors un convexe de petit volume qui contient une proportion de points anormalement élevée, ou, au contraire un convexe de grand volume qui contient une proportion de points anormalement faible par rapport à sa mesure de Lebesgue : la discrépance isotrope est alors élevée !

Cette définition de la discrépance isotrope peut alors être simplifiée en ne considérant que l'ensemble des rectangles dans les convexes. Notons $I^{s,r}$ cet ensemble. Nous pouvons alors définir la discrépance extrême.

**La discrépance extrême :** la discrépance extrême d'une suite $(x_n, n > 0)$ est définie sur tous les intervalles du cube unité par :

$$D_n(x) = \sup_{P \in I^{s,r}} \left| \frac{\#(P, (x_n))}{n} - \lambda(P) \right|.$$

La figure 2.4 illustre l'estimation de cette discrépance. En élargissant l'ensemble des hyperrectangles à l'ensemble des convexes de $I^s$, nous pouvons définir la discrépance isotrope.

Cette définition peut encore être simplifiée en ne considérant parmi les rectangles que ceux qui sont ancrés à l'origine. Nous définissons alors la discrépance à l'origine, plus couramment appelée discrépance star.

1. Nous entendons par « bien répartis », des points dont la distribution est « uniforme » au sens de la discrépance isotrope.

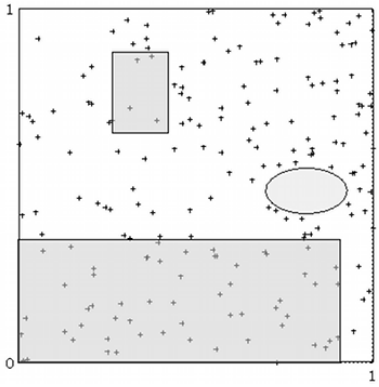



**FIGURE 2.3:** *Estimation de la discrépance isotrope $J_n(x)$ d'une suite x : la discrépance isotrope d'une suite correspond à la mesure de l'écart maximum entre la proportion de points présents dans tout convexe et la proportion du volume du convexe par rapport au volume total du domaine considéré.*

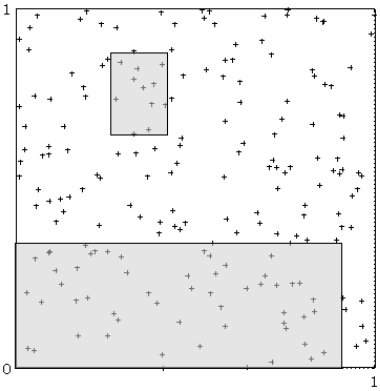



**FIGURE 2.4:** *Estimation de la discrépance extrême $D_n(x)$ d'une suite x : la discrépance extrême d'une suite correspond à la mesure de l'écart maximum entre la proportion de points présents dans tout rectangle et la proportion du volume du rectangle par rapport au volume total du domaine considéré.*

**Discrépance à l'origine ou discrépance star :** définissons $I^{s*}$ comme l'ensemble des hyper-rectangles de $I^s$ contenant l'origine, *i.e.* que ces éléments sont de la forme $\Pi_{i=1}^{s}[0,u_i)$. La discrépance star ou discrépance à l'origine est alors définie par :

$$D_n^*(x) = D^*(x_{(n)}) = \sup_{P \in I^{s*}} \left| \frac{\#(P, x_{(n)})}{n} - \lambda(P) \right|. \tag{2.2}$$

La détermination de cette quantité revient donc à chercher l'intervalle ancré en zéro qui contient la densité la plus anormalement faible ou élevée par rapport à son volume. Nous avons la relation : $0 \leq D_n^*(x) \leq 1$ et plus la discrépance est faible, plus la suite est uniforme, et inversement.
Autrement dit, la discrépance star ou discrépance à l'origine d'une suite correspond à la mesure de l'écart entre la proportion de points présents dans un hyper-rectangle ancré à l'origine et la proportion attendue par rapport au volume total sous l'hypothèse d'une distribution « uniformément » répartie. Un schéma représentatif de cette discrépance est présenté à la figure 2.5.

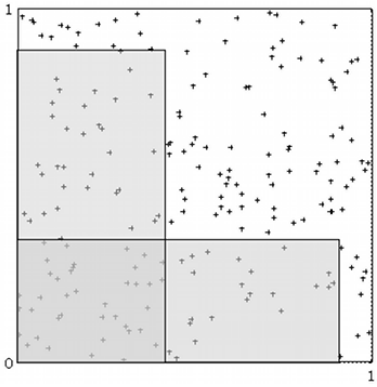



**FIGURE 2.5:** *Estimation de la discrépance star $D_n^*(x)$ d'une suite x : la discrépance star ou discrépance à l'origine d'une suite correspond à la mesure de l'écart entre la proportion de points présents dans un hyper-rectangle ancré à l'origine et la proportion du volume de l'hyper-rectangle par rapport au volume total du domaine considéré.*

La discrépance star peut s'interpréter également comme la norme $L^\infty$ de la fonction *discrépance locale* qui associe à tout intervalle $P \in I^{s*}$ la valeur $\frac{\#(P,x(n))}{n} - \lambda(P)$.

**Cas des suites uniformément réparties :** notons $\hat{F}_n$ la fonction de répartition empirique associée à un échantillon $X_1,\ldots,X_n$ de variables aléatoires. Celle-ci est définie par $\hat{F}_n(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[0,t]}(X_i)$, $\forall t \in [0,1]^s$. Notons $U$ la fonction de répartition de la loi uniforme. Dans le cas d'une suite de variables $X_1,\ldots,X_n$ de variables aléatoires indépendantes et identiquement distribuées de loi uniforme, la discrépance star peut s'écrire sous la forme :

$$D_n^*(x) = \left\| \hat{F}_n - U \right\|_{L^\infty(\mathscr{X})}.$$

Cela représente la norme $L^\infty$ de la fonction écart entre la fonction de répartition empirique et celle d'une loi uniforme. Statistiquement, à un coefficient multiplicatif près dépendant de $n$, il s'agit de la statistique de test d'un test d'uniformité de Kolmogorov-Smirnov.

**Discrépance d'ordre $q$ :** de la même façon que pour la discrépance star, en utilisant une norme d'un espace fonctionnel, on peut définir la discrépance d'ordre $q$ par :

$$T_n^{*,q}(x) = \left[ \int_{P \in I^{s*}} \left( \frac{\#(P,x(n))}{n} - \lambda(P) \right)^q dP \right]^{\frac{1}{q}}.$$

Il s'agit de la norme de la fonction de discrépance locale dans l'espace fonctionnel $L^q$. Dans le cadre de $q = 2$, on parle de discrépance carrée moyenne et on la note $T_n^*$.

D'une manière générale, il est possible d'utiliser différentes normes pour un espace fonctionnel. Par conséquent, il est également possible de définir d'autres discrépances... De plus, il est également possible d'utiliser des espaces de convexes utilisant des combinaisons de rectangles et de définir d'autres discrépances. Niederreiter et Spanier (1998) définissent toutes ces discrépances. Citons par exemple la discrépance modifiée, la discrépance $L^p$ centrée ou la discrépance $L^p$ symétrique. Ces discrépances n'ayant peu d'intérêt dans notre cadre d'études, elles ne sont pas présentées.

**Équivalence des différentes discrépances :** soit $x = (x_1, \ldots, x_n)$ une suite composée de $n$ points dans $I^s$.

Kuipers et Niederreiter (1974) montrent que l'on a la relation suivante :

$$D_n^*(x) \leq D_n(x) \leq J_n(x). \tag{2.3}$$

Niederreiter (1992) montre que l'on a également les relations suivantes :

$$J_n(x) \leq 4s\left[D_n(x)\right]^{\frac{1}{s}} \tag{2.4}$$

$$D_n(x) \leq 2^s D_n^*(x) \tag{2.5}$$

$$0 < T_n^*(x) \leq D_n^*(x) \leq 1. \tag{2.6}$$

Kuipers et Niederreiter (1974) montrent qu'il y a équivalence entre :

1. la suite $x$ est équirépartie ;
2. $\lim_{n\to+\infty} D_n^*(x) = 0$ ;
3. $\lim_{n\to+\infty} D_n(x) = 0$ ;
4. $\lim_{n\to+\infty} J_n(x) = 0$ ;
5. $\lim_{n\to+\infty} T_n^{*,q}(x) = 0$ pour tout $q$.

Avec les équivalences précédentes, et les inégalités 2.3 à 2.6, nous pouvons en déduire que toutes ces définitions de discrépance sont équivalentes. Dans la suite de cette thèse, sauf mention du contraire, nous ne considérons que la discrépance star d'une suite définie à l'équation 2.2.

### 2.1.2 Estimation des discrépances d'une suite

L'estimation de la discrépance d'une suite est un exercice réputé difficile en général. Nous présentons dans cette section les résultats théoriques en dimension $s = 1$ et $s = 2$, puis nous nous intéressons au cas de la dimension quelconque. Enfin, nous faisons un parallèle entre discrépances et statistiques de test de tests d'adéquation à une loi uniforme.

**Calcul de la discrépance en dimension 1 :** soit $x = (x_1, \ldots, x_n)$ une suite de $[0,1]$. Sans perte de généralité, nous pouvons réorganiser la suite de façon à ce que que $x_1 \leq \cdots \leq x_n$. Niederreiter (1972) montre alors que la discrépance s'exprime de manière suivante :

$$D_n^*(x) = \frac{1}{2n} + \max_{1\leq i<n} \left| x^i - \frac{2i-1}{2n} \right|,$$

$$D_n(x) = \frac{1}{n} + \max_{1\leq i<n} \left( \frac{i}{n} - x^i \right) - \min_{1\leq i<n} \left( \frac{i}{n} - x^i \right).$$

**Calcul de la discrépance en dimension 2 :** soit $x = (x_1, \ldots, x_n)$ une suite de $I^2$ dont les points ont été préalablement triés selon l'ordre croissant de leur première composante. Posons $x_0 = (0,0)$ et $x_{n+1} = (1,1)$. Pour chaque indice $i \in 0, \ldots, n$, on note $\xi_0^i, \ldots, \xi_{i+1}^i$ la séquence obtenue en réorganisant le sous-ensemble de secondes composantes $x_1^2, \ldots, x_i^2, x_{n+1}^2$ de sorte que $0 = \xi_0^i \leq \xi_1^i \leq \cdots \leq \xi_i^i \leq \xi_{i+1}^i = 1$.
Bundschuh et Zhu (1993) ont démontré la formule suivante en dimension 2 :

$$D_n^*(x) = \max_{0 \leq i \leq n} \max_{0 \leq k \leq i} \max \left\{ \left| \frac{k}{n} - x_i^1 \xi_k^i \right|, \left| \frac{k}{n} - x_{i+1}^1 \xi_{k+1}^i \right| \right\}.$$

**Calcul de la discrépance en dimension quelconque :** soit $x = (x_1, \ldots, x_n)$ une suite de dimension $s$ quelconque, alors Warnock (1972) a montré que :

$$(T_n^{*,q}(x))^2 = \frac{1}{3^s} - \frac{1}{n 2^{s-1}} \sum_{i=1}^{n} \prod_{k=1}^{s} \left(1 + x_i^k\right) \left(1 - x_i^k\right) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{k=1}^{s} \left(1 - \max\left(x_i^k, x_j^k\right)\right).$$

Cette formule de Warnock montre que la discrépance carrée moyenne d'une séquence de $n$ points peut être calculée en $\mathcal{O}\left(n^2\right)$ opérations.
Dans le cas de suites particulières de Halton et de Niederreiter en dimensions $s = 2, 3, 4$ et 6 (voir section 2.2 pour une définition de ces suites) Schlier (2004) a estimé les discrépances pour certaines tailles de suites.
Pour une estimation empirique, Niederreiter (1992) a démontré que le problème est discrétisable et peut se résoudre en un nombre fini d'étapes. Malheureusement, la complexité des algorithmes croît exponentiellement avec la dimension. Thiemard (2001) et Tovstik (2007) font une revue des différents algorithmes existants et proposent de nouveaux algorithmes.

**Discrépances et statistiques de test :** notons $\hat{F}_n$ la fonction de répartition empirique associée à un échantillon $X_1, \ldots, X_n$ de variables aléatoires indépendantes et identiquement distribuées. Celle-ci est définie par $\hat{F}_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{[0,t]}(X_i)$, $\forall t \in [0,1]^s$. Notons $U$ la fonction de répartition de la loi uniforme ; *i.e.* $\forall u \in [0,1]^s, U(u) = \Pi_{i=1}^s (u^i)$.

**- Discrépance star et statistique de test de Kolmogorov-Smirnov :** en dimension $s = 1$, la discrépance star peut s'écrire également sous la forme :

$$D_\infty^*(n) = \left\| \hat{F}_n - U \right\|_{L^\infty(\mathcal{X})} = \sup_{x \in \mathcal{X}} \left| \hat{F}_n(x) - U(x) \right|.$$

Cela représente la norme $L^\infty$ de la fonction écart entre la fonction de répartition empirique et celle d'une loi uniforme. Statistiquement, il s'agit de la statistique de test du test statistique d'adéquation de Kolmogorov-Smirnov de l'échantillon à une loi uniforme. Pour tester l'hypothèse, à l'aide d'un test statistique d'adéquation de Kolmogorov-Smirnov, qu'un échantillon donné est issu d'une variable aléatoire uniforme, il suffit de calculer la discrépance star de celui-ci et de comparer la valeur obtenue au fractile d'une loi de Kolmogorov. Massart (1990) donne une majoration de cette loi par :

$$\forall \varepsilon > 0, \mathbb{P}\left[ n^{1/2} D_n^*(x) \geq t \right] \leq 2 \exp\left(-2t^2\right).$$

En pratique, en ayant fixé un risque de première espèce à $\alpha$, si la discrépance de l'échantillon est supérieure au fractile d'ordre $1 - \alpha$ d'une loi de Kolmogorov à $n - 1$ degrés de liberté, alors nous

pouvons rejeter, au risque d'erreur de première espèce de $\alpha$, l'hypothèse selon laquelle l'échantillon est issu d'une distribution uniforme.

En dimension $s \geq 2$ quelconque, Dvoretzky *et al.* (1956) démontrent alors que $\forall \varepsilon > 0$, $\exists C_{\varepsilon,s} > 0$ telle que $\mathbb{P}\left[n^{1/2} D_n^*(x) \geq t\right] \leq C_{\varepsilon,s} \exp\left(-(2-\varepsilon)t^2\right)$. Le choix optimal en dimension 1 de $C_{\varepsilon,1} = 2$ ne s'applique plus. Le calcul de la loi de la statistique de Kolmogorov-Smirnov est alors délicat et une expression exacte de la loi limite n'est pas connue à ce jour.

**- Discrépance d'ordre 2 et statistique de test de Cramer Von Mises :** la discrépance d'ordre 2, $T^{n,2}$, lorsqu'elle est mise au carré peut s'écrire alors sous la forme :

$$\left(T^{n,2}\right)^2(x) = \left\|\hat{F}_n - U\right\|^2_{L^\infty(\mathscr{X})} = \sup_{x \in \mathscr{X}} \left|\hat{F}_n(x) - U(x)\right|^2.$$

Notons la variable aléatoire $W_n = n\left(T^{n,2}\right)^2(x) = n\left\|\hat{F}_n - U\right\|^2_{L^\infty(\mathscr{X})} = n \sup_{x \in \mathscr{X}} \left|\hat{F}_n(x) - U(x)\right|^2$. Cette variable $W_n$ correspond à la statistique du test statistique d'adéquation de Cramer Von Mises d'une loi uniforme. Statistiquement, cette discrépance au carré correspond donc, à un coefficient multiplicatif $n$ près, à la statistique de test du test statistique d'adéquation de Cramer Von Mises d'une loi uniforme. Deheuvels *et al.* (2006) démontrent que $W_n$ converge vers un pont brownien standard multivarié, *i.e.* converge en loi vers une somme pondérée de variables aléatoires de loi du Khi2. Cette statistique admet donc une loi limite permettant de déterminer une valeur définissant, pour un risque de première espèce fixé, une zone de rejet de l'hypothèse selon laquelle l'échantillon est issu d'une distribution aléatoire uniforme.

**- Différence entre le test d'adéquation de Kolmogorov et celui de Cramer Von Mises :** le test de Cramer Von Mises possède les mêmes applications que le test de Kolmogorov. La différence entre ces deux tests réside dans le fait que pour le test de Kolmogorov seul l'écart maximum entre la distribution empirique et la distribution théorique entre en considération alors que la statistique du test de Cramer Von Mises prend mieux en compte l'ensemble des données parce que la somme des écarts intervient. Le test de Kolmogorov est donc beaucoup plus sensible à l'existence de points aberrants dans un échantillon que le test de Cramer Von Mises. On conjecture que le test de Cramer Von Mises est plus puissant, mais cela n'a pas été encore prouvé théoriquement.

**Conclusion sur la discrépance d'une suite de points :** l'estimation de la discrépance d'une suite de points est complexe en termes de nombre de calculs nécessaires. C'est pourquoi, en général, elle n'est jamais estimée : les utilisateurs essayent de travailler avec des suites ayant de « bonnes » discrépances. Nous présentons certaines de ces suites dans la section 2.2. La discrépance d'une suite en tant que telle est surtout utilisée pour des travaux théoriques.

### 2.1.3 Discrépances de suites et suites à faible discrépance

Dans cette partie, nous nous intéressons aux valeurs numériques des discrépances de suites et notamment à la discrépance des suites aléatoires uniformes. A partir de cette dernière nous définissons les suites dites à discrépance faible.

Niederreiter (1992) démontre les relations suivantes pour toute suite $x_{(n)} \in I^s$ de taille $n$ finie :

$$D_n^*(x_1,\ldots,x_n) \geq \frac{1}{2n} \text{ et } \frac{1}{n} \leq D_N(x_1,\ldots,x_n), \text{pour } s = 1. \tag{2.7}$$

Il démontre également que :

$$D_n^*(x) \geq 0.0233.... \frac{\log(n)}{n}, \text{ pour } s = 2. \tag{2.8}$$

Roth (1954) montre que, pour $n$ fini, il existe une constante $C_s > 0$ dépendant uniquement de la dimension $s$ telle que, pour toute séquence $x = \{x_1, \ldots, x_n\}$, on a l'inégalité :

$$D_n^*(x) \geq C_s \frac{(\log(n))^{\frac{s-1}{2}}}{n}.$$

En généralisant ce résultat, il obtient que pour toute dimension $s$ , il existe une constante $C_s'$ ne dépendant que de la dimension $s$ telle que pour toute séquence $x$ de taille infinie, *i.e.* $x = \{x_1, x_2, \ldots, x_n, \ldots\}$, on ait l'égalité suivante :

$$D_n^*(x) \geq C_s' \frac{(\log(n))^{\frac{s}{2}}}{n} \text{ pour une infinité de valeurs de } n.$$

**Discrépance d'une suite aléatoire uniforme :** soit $x_{(n)} = (x_1, \ldots, x_n)$ une réalisation issue d'une loi aléatoire uniforme sur $I^s$. Kiefer (1961) montre alors que :

$$\limsup_{n \to +\infty} \sqrt{\frac{2n}{\ln(\ln(n))}} D^*(x_{(n)}) = 1 \text{ presque sûrement.}$$

Il est intéressant de remarquer que la dimension de l'espace n'intervient pas dans ce résultat. Celui-ci est important car il sert de base à la classification des suites entre elles et à la définition suivante des suites à discrépance faible.

**Suites à discrépance faible :** on définit les suites à discrépance faible par des suites dont la discrépance star est asymptotiquement meilleure que celle d'une suite aléatoire.

**Discrépance d'une grille régulière :** Niederreiter (1992) montre que la discrépance d'une grille régulière à $n$ points en dimension $s$ se situe en $\mathcal{O}\left(\frac{1}{\sqrt[s]{n}}\right)$. Par conséquent, une grille régulière n'est pas à discrépance faible.

Actuellement, on ne connaît pas de suites pour lesquelles $D_n^*(x) = \mathcal{O}\left(\frac{(\log(n))^{\frac{s}{2}}}{n}\right)$ et il est conjecturé qu'il n'en existe vraisemblablement pas. On sait construire des suites pour lesquelles $\mathcal{O}\left(\frac{(\log(n))^s}{n}\right)$ ou $\mathcal{O}\left(\frac{(\log(n))^{s-1}}{n}\right)$. On conjecture que cet ordre de discrépance est l'ordre exact et qu'il n'existe aucune suite présentant une décroissance plus rapide. La section suivante s'intéresse à la construction de quelques suites à discrépance faible.

## 2.2 Générer des suites à discrépance faible

Pour générer une suite à faible discrépance, il faut éviter au maximum le regroupement de points dans une région donnée, phénomène que l'on trouve souvent avec la méthode de Monte Carlo (dû à l'existence de corrélations entre les dimensions lorsque la dimension totale est grande). Nous allons présenter dans cette section des suites classiques à faible discrépance, *i.e.* dont la discrépance

est de l'ordre de $\mathscr{O}\left(\frac{(\log(n))^s}{n}\right)$. Nous commençons par la suite de Van Der Corput en dimension $s = 1$, puis nous généralisons cette suite en dimension supérieure avec les suites de Halton, Faure et Hammersley. Nous présentons ensuite la suite de Sobol, avant de classifier toutes ces suites dans la famille des suites de Niederreiter. Enfin, nous présentons les suites de $\{n\alpha\}$.

Cette section présente un état de l'art des méthodes actuelles de génération des suites à discrépance faible. De nature assez technique, elle peut être ignorée en première lecture ou si l'intérêt du lecteur est plus porté sur l'utilisation de ces suites en apprentissage plutôt que sur leur génération.

### 2.2.1 La suite de Van Der Corput

La philosophie de la suite de Van Der Corput (voir Van der Corput (1935)) consiste à diviser le volume d'intégration en sous-volumes et placer un point dans chacun de ceux-ci. Cette opération étant réalisée, on passe alors à plus de points en divisant les nouveaux volumes en plus petits.
Mathématiquement, elle se base sur la décomposition $p$-adique des nombres entiers : soit $p$ un nombre supérieur à 1. Tout entier positif $n$ peut se décomposer en une combinaison linéaire de puissances de $p$ à coefficients entiers, i.e. $n = a_0 + a_1 p + \cdots + a_r p^r$ et $0 \leq a_i < p$ où $0 \leq a_i \leq r$.
La suite de Van Der Corput $\left(x_{(n)}\right)$ est alors définie par :

$$x_n = \Phi_x(n) = \sum_{i=0}^{r} \frac{a_i}{p^{i+1}}.$$

Exemple de calcul : calcul du 25$^{\text{ième}}$ terme d'une suite de base $p = 3$

$25 = 1*3^0 + 2*3^1 + 2*3^2$ et donc $p_{25} = \Phi_3(25) = \sum_{i=0}^{2} \frac{a_i}{3^{i+1}} = \frac{1}{3} + \frac{2}{3^2} + \frac{2}{3^3} = \frac{17}{27}$.

Le tableau 2.1 fournit les neufs premiers termes de la suite de Van Der Corput en base 2, 3, 5 et 7. Une représentation graphique de cette suite est réalisée à la figure 2.6. On peut remarquer que la suite commence à déposer des points dans l'intervalle [0,1] (en base 2, les 3 premiers termes de la suite), puis redécoupe cet intervalle (termes 4 à 7 de la suite), etc .... Il en ressort un alignement régulier au cours des itérations des points de la suite.

| Terme | Base 2 | Base 3 | Base 5 | Base 7 |
|---|---|---|---|---|
| 1 | 1/2 | 1/3 | 1/5 | 1/7 |
| 2 | 1/4 | 2/3 | 2/5 | 2/7 |
| 3 | 3/4 | 1/9 | 3/5 | 3/7 |
| 4 | 1/8 | 4/9 | 4/5 | 4/7 |
| 5 | 5/8 | 7/9 | 1/25 | 5/7 |
| 6 | 3/8 | 2/9 | 6/25 | 6/7 |
| 7 | 7/8 | 5/9 | 11/25 | 1/49 |
| 8 | 1/16 | 8/9 | 16/25 | 8/49 |
| 9 | 3/16 | 1/27 | 21/25 | 15/49 |

**TABLE 2.1:** *Neuf premiers termes de la suite de Van Der Corput en base 2, 3, 5 et 7.*

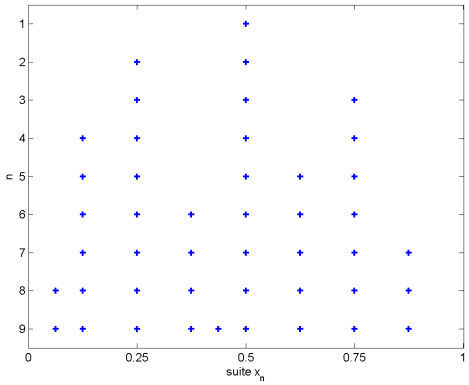



**FIGURE 2.6:** *Représentation de la construction des 8 premiers termes d'une suite de Van Der Corput en base 2. On peut remarquer un alignement régulier au cours des itérations des points de la suite.*

Il est possible d'obtenir des suites de Van Der Corput ayant une discrépance asymptotiquement plus faible que celle des suites standards de Van Der Corput présentées ci-dessus. Pour cela, il suffit d'introduire une permutation des éléments de l'ensemble $P = \{1,2,\ldots,p-1\}$. Notons $\sigma$ une permutation définie sur cet ensemble $P$. Les éléments de la suite sont alors de la forme : $x_n = \Phi_p(n) = \sum_{i=0}^{r} \frac{\sigma(a_i)}{p^{i+1}}$.

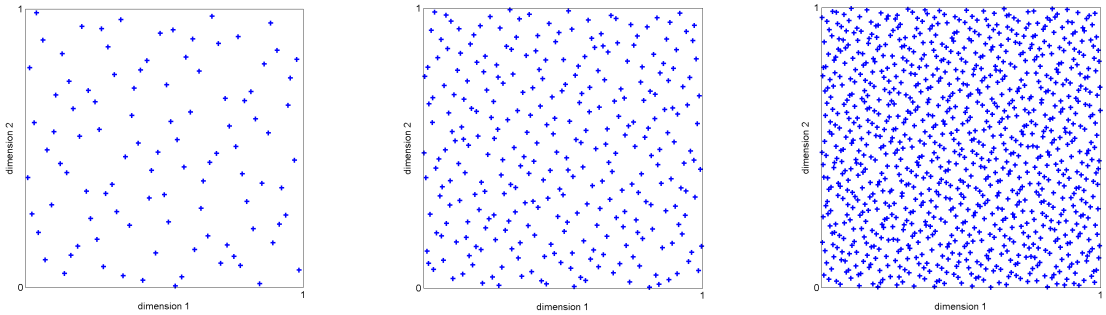
### 2.2.2 La suite de Halton

La suite de Halton, définie par Halton (1960), est une généralisation de la suite de Van Der Corput en dimension $s > 1$. L'idée est de générer des suites de Van Der Corput indépendamment dans chaque dimension. Afin d'éviter l'alignement que cela produirait (toutes les coordonnées seraient les mêmes), les différentes composantes sont calculées dans différentes bases. Mathématiquement, cela se définit ainsi :
soient $p_1,\ldots,p_s$ des entiers supérieurs à 1 et premiers entre eux deux à deux, la suite de Halton est définie par :

$$x_n = (\Phi_{p_1}(n),\ldots,\Phi_{p_s}(n))$$

où les fonctions $\Phi$ sont les fonctions de base de la suite de Van Der Corput.

Une représentation graphique des suites de Halton en dimension 2 de taille est donnée à la figure 2.7. On peut remarquer que si l'on projette les points de la suite sur les axes de chaque dimension, on obtient une multitude de valeurs de projection. Cette multitude de valeurs est supérieure à l'ensemble des valeurs que l'on pourrait obtenir en projetant une grille régulière. Ceci est une caractérisation graphique des suites à faible discrépance.




(a) Suite de 100 points. (b) Suite de 300 points. (c) Suite de 1 000 points.

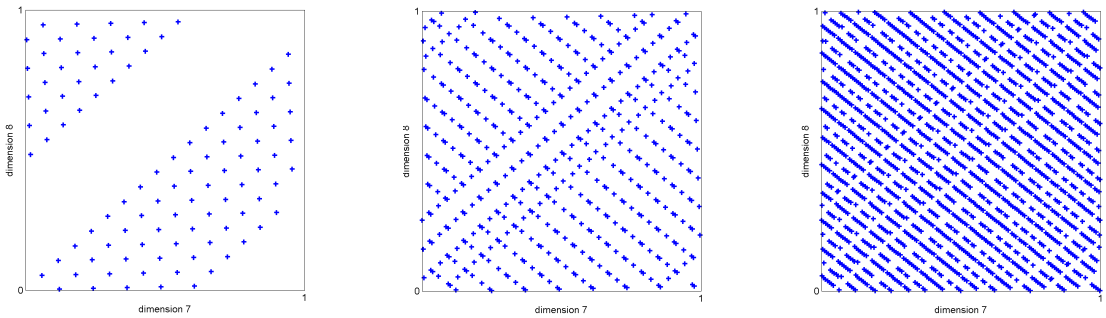
**FIGURE 2.7:** *Suite à discrépance faible de Halton en dimension 2.*

Pour une suite $x$ de Halton à $n$ éléments, la discrépance vérifie la propriété suivante :

$$D_n^*(x) \leq \frac{s}{n} + \frac{1}{n}\prod_{j=1}^{s}\left(\frac{p_j - 1}{2\log(p_j)}log(n) + \frac{p_j + 1}{2}\right)$$

Le fait de choisir comme bases $p_1, \ldots, p_s$, les $s$ premiers nombres permet de minimiser la constante $\prod_{j=1}^{s}\frac{p_j-1}{2\log(p_j)}$ du terme dominant la majoration.

Cependant, un des inconvénients de cette suite est qu'elle ne possède pas de « bonnes » propriétés de recouvrement du carré pour des dimensions élevées : elle nécessite un nombre important de points. Cela s'illustre parfaitement à la figure 2.8 où l'on peut remarquer qu'en dimension 8, il existe des directions d'alignement des points et que pour éviter ces alignements il faut un nombre important de points.




(a) Coupe d'une suite de 100 points. (b) Coupe d'une suite de 500 points. (c) Coupe d'une suite de 2000 points.

**FIGURE 2.8:** *Coupe en dimensions 7 et 8 d'une suite à discrépance faible de Halton en dimension 8. On peut remarquer que le remplissage de l'espace n'est pas très bon lorsque la dimension augmente. Pour améliorer le remplissage, il est nécessaire d'augmenter significativement la taille de la suite, mais des alignements apparaissent.*

### 2.2.3 La suite de Faure

Dans la suite de Halton, le mauvais remplissage des points lorsque la dimension augmente est dû au fait que le nombre premier de la base augmente en même temps que la dimension, et remplir correctement l'espace dans une grande dimension nécessite alors de plus en plus de points. Afin

de s'affranchir de cette contrainte, nous pouvons générer la suite de Halton en utilisant le même nombre premier pour chaque dimension. Cette solution permet d'éviter un mauvais remplissage en grande dimension, mais induit une régularité des points : ils ont tous les mêmes valeurs dans les dimensions. Afin d'éviter ceci, on peut permuter les éléments et les dimensions de la suite.

La suite obtenue avec les modifications présentées ci-dessus s'appelle la suite de Faure (voir Faure (1982)). Mathématiquement, elle s'obtient de la manière suivante :

1. décomposer l'indice $n$ sous sa forme $p$-adique :

$$n = a_0 \frac{1}{p} + a_1 \frac{1}{p^2} + \cdots + a_r \frac{1}{p^r} = \sum_{i=0}^{r} \frac{a_i}{p^{i+1}}.$$

2. soit l'application $T_p : \mathbb{N} \to \mathbb{N}$ définie par :

$$T_p(n) = T_p\left(\sum_{i=0}^{r} \frac{a_i}{p^{i+1}}\right) = \sum_{i=0}^{r} \left[\sum_{j=1}^{r} C_i^j a_j mod(p)\right] \frac{1}{p^{i+1}}.$$

3. le $n^{\text{ième}}$ élément de la suite de Faure en dimension $s$ s'obtient ensuite de la manière suivante :

$$x_n = \left(\Phi(n-1), T_p(\Phi(n-1)), \ldots, T_p^{s-1}(\Phi(n-1))\right).$$

La discrépance de la suite de Faure est de l'ordre de $\mathscr{O}\left(\frac{\log^s(n)}{n}\right)$. Faure (1982) montre que la discrépance d'une suite de base $p$ peut être calculée avec les équations suivantes :

- si $s = 2, D_n^*(x) \leq \frac{3}{16(\log(2))^2} \frac{1}{n} (\log(n))^2 + \mathscr{O}\left(\frac{\log(n)}{n}\right)$ ;
- si $s \geq 3, D_n^*(x) \leq \frac{1}{s!} \left(\frac{p-1}{2\log(p)}\right)^s \frac{1}{n} (\log(n))^s + \mathscr{O}\left(\frac{(\log(n))^{s-1}}{n}\right)$.

### 2.2.4 La suite de Hammersley et les suites à faible discrépance aléatoires

La suite de Hammersley, définie par Hammersley (1960), est une suite à faible discrépance en dimension $s$ construite à partir d'une suite de Halton à faible discrépance de dimension $s-1$. Tout comme la suite de Halton, elle se base sur $p_1, \ldots, p_{s-1}$, entiers positifs premiers entre eux deux à deux, et la $s^{\text{ième}}$ dimension est rajoutée régulièrement. Pour générer une suite en dimension $s$ à $n$ termes, le $i^{\text{ième}}$ élément de la suite est défini par :

$$x_i = \left(\frac{i}{n}, \Phi_{p_1}(i), \ldots, \Phi_{p_{s-1}}(i)\right).$$

Hammersley (1960) montre alors que pour une suite $x$ de Hammersley de dimension $s$ à $n$ termes, la discrépance se borne par :

$$D_n^*(x) \leq \frac{s}{n} + \frac{1}{n} \prod_{i=1}^{s-1} \left(\frac{p_i - 1}{2\log(p_i)} \log(n) + \frac{p_i + 1}{2}\right).$$

Dans le cas où $s = 2$, et si la suite de Hammersley est de base $p$ dans les deux dimensions de taille $n = p^m$, nous avons les propriétés suivantes :

1. pour $p$ impair et $m \geq 2$,
$D_{p^m}^*(x) = \frac{p-1}{4p^m} + \frac{1}{p^m}\left(\frac{5}{4} + \frac{1}{p}\right) - \frac{1}{4p^{2m}}$,

2. pour $p$ pair et $m \geq 2$ pair,
$$D^*_{p^m}(x) = \frac{p^2 m}{4p^m(p+1)} + \frac{1}{p^m}\left(\frac{5}{4} + \frac{2p+3}{4(p+1)^2}\right) - \frac{1}{4p^{2m}}\left(1 + \frac{2p+3}{(b+1)^2}\right),$$
3. pour $p$ pair et $m \geq 3$ impair,
$$D^*_{p^m}(x) = \frac{p^2 m}{4p^m(p+1)} + \frac{1}{p^m}\left(\frac{5}{4} + \frac{5p+4}{4p(p+1)^2}\right) - \frac{1}{p^{2m}}\left(\frac{p}{2} - \frac{1}{4} - \frac{1}{p} + \frac{5}{4p^2} - \frac{6p+5}{4p^2(b+1)^2}\right).$$

Le passage d'une suite de Halton à une suite de Hammersley est un principe général. En utilisant la même technique, il est possible de passer d'une suite à faible discrépance de $I^{s-1}$ à une séquence de $n$ points dans $I^s$ dont la discrépance est en $\mathcal{O}\left(\frac{(\log(n))^{s-1}}{n}\right)$. Ce taux est asymptotiquement meilleur, cependant les séquences obtenues ne peuvent pas être facilement étendues à des dimensions encore supérieures sans mettre en péril cette propriété. L'utilisation d'une telle suite n'est donc pas envisageable dans le cas où l'on ne connaît pas le nombre de points à générer.

**Suites de Quasi Monte-Carlo aléatoires :** la technique de la suite de Hammersley fonctionne correctement lorsqu'il s'agit de passer d'une suite à faible discrépance de dimension $s$ à une suite à faible discrépance de dimension $s+1$. Cependant, pour passer à des suites de dimension supérieures, la généralisation est difficile sans mettre en péril la faible discrépance. Une méthode plus poussée réside dans les suites de Halton randomisées comme le définissent Wang et Hickernell (2000) qui introduisent de l'aléatoire dans plusieurs dimensions. A partir de ces travaux, seuls Tuffin (2004), Faure et Lemieux (2008), Okten et Willyard (2008) et Okten (2009) et De Rainville *et al.* (2009) se sont intéressés théoriquement à ces suites et à leurs implémentations informatiques.

**Utilisation des suites de Quasi Monte-Carlo aléatoires :** sur l'utilisation de ces suites sur des problèmes mathématiques on peut citer les travaux de Nguyen *et al.* (2007) en optimisation, de L'Ecuyer *et al.* (2008) en simulations numériques, de Neddermeyer (2011) en estimation de fonctions de densité et de Munger *et al.* (2012) en estimation de fonctions de vraisemblance. Sur l'utilisation de ces suites sur des problèmes « « concrets » non mathématiques », on peut citer les travaux de Tan et Boyle (2000) en économie et de Okten et Eastman (2004) ou de Lemieux (2004) en finance.

**Conclusion sur les suites aléatoires généralisées :** la bibliographie théorique sur les suites de Quasi Monte-Carlo aléatoires est très faible sur une période de presque dix ans, tout comme la bibliographie relative aux applications de cette méthode. De plus, peu de scientifiques ont étudié ce sujet : la majeure partie des avancées sont cosignées par Okten. Cela montre bien, pour nous, que cette approche ne permet de s'affranchir que temporairement du problème.

### 2.2.5 La suite de Sobol

Les suites de Sobol, proposées par Sobol (1967) se définissent à partir de récurrences linéaires en arithmétique modulo 2 sur l'ensemble fini $E = \{0,1\}$ et de polynômes primitifs.

**Polynômes primitifs :** un polynôme de la forme $t^m + u_1 t^{m-1} + \cdots + u_{m-1}t + u_m$ de degré $m$ est dit primitif sur le corps $E$ s'il est irréductible sur $E$ et si le plus petit entier $i$ pour lequel il divise $t^i + 1$ est égal à $2^m - 1$. Il n'existe pas d'algorithmes capables de générer ces polynômes. Néanmoins, Niederreiter (1992) a calculé les premiers polynômes et a réalisé une table. Ces polynômes de degré inférieur ou égal à 5 sont présentés à la table 2.2.

| | | |
|---|---|---|
| $t+1$ | $t^2+t+1$ | $t^3+t+1$ |
| $t^3+t^2+1$ | $t^4+t+1$ | $t^4+t^3+1$ |
| $t^5+t^2+1$ | $t^5+t^3+1$ | $t^5+t^3+t^2+t+1$ |
| $t^5+t^4+t^2+t+1$ | $t^5+t^4+t^3+t+1$ | $t^5+t^4+t^3+t^2+1$ |

**TABLE 2.2:** *Polynômes primitifs de degré inférieur ou égal à 5.*

Notons $\oplus$ l'opérateur « ou exclusif » en logique. A partir d'un polynôme primitif $t^m+u_1t^{m-1}+\cdots+u_{m-1}t+u_m$ de degré $m$ sur $E$, d'un ensemble d'entier $\{l_1,\ldots,l_m\}$ tels que $1\leq l^i<2^i$ pour tout $i\in\{1,\ldots,m\}$, on définit l'ensemble $\{l_{m+1},l_{m+2}\ldots\}$ en utilisant la relation de récurrence :

$$l_i=2u_1l_{i-1}\oplus 4u_2l_{i-2}\oplus\cdots\oplus 2^{d-1}u_{d-1}l_{i-d+1}\oplus\left(2^dl_{i-d}\oplus l_{i-d}\right).$$

**Suite de Sobol en dimension 1 :** la suite $S=\{x_0,x_1,\ldots\}\in I^1$ est définie par :

$$x_i=\frac{1}{2^m}\left(\oplus_{k=1}^m c_k(i)l_k\right),$$

où $(c_1(i),\ldots,c_m(i))$ est la représentation binaire de $i$

$$i=\sum_{k=1}^m c_k(i)2^{k-1}\ avec\ m=\begin{cases}1 & pour\ i=0\\ 1+\lfloor log_2(i)\rfloor & sinon\end{cases}.$$

**Exemple :** soit $t^3+t+1$ un polynôme primitif de degré d =3. Les nombres $l_1,\ldots,l_d$ doivent être impairs et satisfaire la contrainte $l_i<2^i$.
On choisit par exemple $l_1=1,l_2=3\ et\ l_3=7$. La relation de récurrence imposée sur les $l_i$ est :

$$l_i=4l_{i-2}\oplus(8l_{i-3}\oplus l_{i-3}),\ pour\ tout\ i>3.$$

On obtient donc $l_4$ de la manière suivante :

$$\begin{aligned}l_4 &= 4l_2\oplus 8l_1\oplus l_1\\ &= 1100\oplus 1000\oplus 0001 \text{ en binaire}\\ &= 0101 \text{ en binaire}\\ &= 5\end{aligned}$$

Calculons par exemple le $13^{\text{ième}}$ (1101 en binaire) point de cette suite de Sobol en dimension 1 :

$$\begin{aligned}x_{13} &= \frac{1}{16}(l_1\oplus l_3\oplus l_4)\\ &= \frac{1}{16}(0001\oplus 0111\oplus 0101)\\ &= \frac{3}{16}\end{aligned}$$

**Suite de Sobol en dimension $s$ :** pour obtenir une suite de Sobol en dimension $s$, il suffit de choisir $s$ polynômes primitifs distincts et de juxtaposer les suites correspondantes en dimension 1.

Sobol (1967) montre que pour une suite de Sobol $x$ de $n$ termes en dimension $s$, on a

$$D_n^*(x) = \frac{2^{t_s}}{s!\,(\log(2))^s}\frac{(\log(n))^s}{n} + \mathcal{O}\left(\frac{(log(n))^{s-1}}{n}\right)$$

où $t_s$ ne peut être majorée par une fonction linéaire de la dimension :

$$k\frac{s\log(s)}{\log(\log(s))} \leq t_s \leq \frac{s\log(s)}{log(2)} + \mathcal{O}(s\log(\log(s)))$$

### 2.2.6 Les suites de Niederreiter, *alias* les $(t,s)$-suites et les $(t,m,s)$-réseaux et la classification des suites

La notion de $(t,s)$-suites et les $(t,m,s)$-réseaux est due à Niederreiter (1992) qui a essayé de classifier et de généraliser les constructions des suites de Van Der Corput et de Halton... Ces suites sont également appelées les suites de Niederreiter dans la littérature.
Un $(t,m,s)$-réseau en base $b$ est un ensemble de $b^m$ points dans le cube unité $I^s$, pour lequel la discrépance locale (voir paragraphe 2.1.1 page 30) est nulle pour une certaine famille de $I^s$. Une $(t,s)$-suite en base $b$ est une suite dont certains segments de longueur $b^m$ (avec $m \geq t$) sont des $(t,m,s)$-réseaux en base $b$.

Un intervalle élémentaire en base $b$ est un intervalle de la forme :

$$\prod_{i=1}^{s}\left[\frac{a_j}{b^{e_j}}, \frac{a_j+1}{b^{e_j}}\right] \text{ où } a_j, b_j \in \mathbb{N} \text{ et } a_j < b^{e_j} \text{ pour tout } j \in \{1,\ldots,s\}.$$

Soit une paire d'entiers $0 \leq t \leq m$. Un $(t,m,s)$-réseau en base $b$ est une séquence $x$ de $b^m$ points de $I^s$ telle que $\#(P,x) = b^t$ pour tout intervalle élémentaire $P$ en base $b$ de volume $\lambda(P) = b^{t-m}$.
Soit un entier $t \geq 0$. Une $(t,s)-$suite en base $b$ est une suite de points $(x_0, x_1, \ldots)$ telle que pour toute paire d'entiers $k \geq 0$ et $\geq t$, la séquence $\left(x_{kb^m}, \ldots, x_{(k+1)b^m-1}\right)$ est un $(t,m,s)$-réseau en base $b$.

La notion de réseau est importante en pratique car elle fournit des garanties de bonne répartition pour un nombre fini de points. En effet, pour $m, s,$ et $b$ fixés, plus la valeur de $t$ est petite, meilleures sont les propriétés d'un $(t,m,s)$-réseau en base $b$.

Les suites de van der Corput en base $b$ sont des $(0,1)$-suites en base $b$, les suites de Sobol sont des $(t,s)$-suites en base 2 et les suites de Faure en base $b$ sont des $(0,s)$-suites en base $b$. Par contre, les suites de Halton ne sont pas des $(t,s)-$suites, car elles ne sont pas construites à partir d'une base unique, mais elles possèdent des propriétés similaires.

### 2.2.7 Les suites $\{n\alpha\}$

Soit $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_s) \in \mathbb{R}^s$ tel que $1, \alpha_1, \alpha_2, \ldots, \alpha_s$ soient linéairement indépendants sur $\mathbb{Q}$. On peut choisir par exemple $\alpha = \left(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_s}\right)$ où les $p_i$ sont les $s$ premiers nombres premiers. Les suites $\{n\alpha\}$ sont de la forme :

$$x_n = (\{n\alpha_1\}, \{n\alpha_2\}, \ldots, \{n\alpha_s\})$$

où $\{x\}$ est la partie fractionnaire du nombre $x$.

Pour tout $\varepsilon > 0$, la discrépance d'une suite $\{n\alpha\}$ est en $\mathcal{O}\left(\frac{(\log(n))^{1+s+\varepsilon}}{n}\right)$ pour presque tout $\alpha \in \mathbb{R}^s$. Enfin, aucune valeur numérique de $\alpha$ fournissant une discrépance de l'ordre de $\mathcal{O}\left(\frac{(\log(n))^{1+s}}{n}\right)$ n'est connue en dimension $s \geq 2$.

## 2.3 Comportement de la discrépance avec la dimension

Nous avons défini les suites à discrépance faible à la section 2.1.3 comme étant des suites dont la discrépance star est asymptotiquement meilleure que celle d'une suite aléatoire uniforme. Par l'inégalité 2.1.3 (page 34), nous savons que la discrépance d'une suite aléatoire uniforme de taille $n$ est de l'ordre de $\mathcal{O}\left(\sqrt{\frac{\ln(\ln(n))}{2n}}\right)$, indépendamment de la dimension.

**Vitesse de convergence de la discrépance des suites à discrépance faible :** nous avons pu voir que les suites à discrépance faible s'implémentent facilement en informatique et que l'obtention de points est rapide. Les résultats théoriques sur la majoration de la discrépance star font apparaître des constantes $C_s$ dont il est important de contrôler l'évolution quand la dimension augmente. Actuellement les suites à discrépance faible utilisées ont une majoration de la discrépance de $C_s\frac{(\log(n))^s}{n}+\mathcal{O}\left(\frac{(\log(n))^{s-1}}{n}\right)$, décroissant avec une vitesse en $\frac{(\log(n))^s}{n}$. En dérivant ce terme, on peut se rendre compte que $\frac{(\log(n))^s}{n}$ est croissant pour $n\in[0,\exp(s)]$. Cette évolution n'est pas en adéquation avec l'idée que plus on a de points, plus la discrépance est petite. Par conséquent, ce majorant ne semble pas bon tant que le nombre de points est inférieur à $\exp(s)$.
Quand la dimension augmente, ce nombre devient vite inaccessible : le tableau 2.3 présente les valeurs minimales des tailles de suites à partir desquelles la dispersion peut théoriquement diminuer. Celles-ci croissent de manière exponentielle !

| s | 2 | 4 | 6 | 7 | 8 | 9 | 12 | 15 | 20 |
|---|---|---|---|---|---|---|---|---|---|
| $\exp(s)$ | 7,39 | 54,60 | 403,4 | 1 096,6 | 2 981 | 8 103,1 | $1,6.10^5$ | $3,2.10^6$ | $4,89.10^8$ |

**TABLE 2.3:** *Tailles minimales des suites à partir desquelles la majoration de la discrépance diminue.*

La figure 2.9 représente l'évolution de la borne de la discrépance en fonction des tailles des suites et des dimensions du problème. La courbe majorante en bleu représente l'évolution de l'ordre de la discrépance d'une suite aléatoire uniforme, *i.e.* que son ordre de grandeur est en $\mathcal{O}\left(\sqrt{\frac{\ln(\ln(n))}{2n}}\right)$.
Les autres courbes rouge, bleue et verte représentent l'évolution de cet ordre pour les dimensions 2,3 et 4. On peut remarquer que ces ordres de grandeur sont inférieurs à celui de la suite aléatoire. Il est également intéressant de remarquer que ces courbes ont une tendance à se rapprocher de la courbe de la distribution aléatoire lorsque la dimension augmente. Enfin, la courbe noire représente l'ordre de grandeur pour une suite en dimension 5. (A cause de la constante $C_s$ élevée, cette courbe a été réduite en ordonnées afin que le graphique soit lisible : cela ne change en rien l'interprétation des résultats qui, comme pour la définition de la discrépance faible, sont des résultats asymptotiques). Nous pouvons remarquer encore cette convergence de la courbe vers celle de l'aléatoire. Cela illustre bien le fait que, lorsque la dimension augmente, le comportement des suites à faible discrépance se rapproche de celui des suites aléatoires. Autrement dit, les suites à faible discrépance perdent en puissance avec l'augmentation de la dimension.

**Conclusion :** en 1961, Bellman (1961) invente le terme de malédiction de la dimensionnalité, ou fléau de la dimension, (en anglais, *curse of dimensionality*) pour qualifier le problème de l'augmentation explosive du volume de données associée à l'augmentation de dimensions dans un espace mathématique. Le critère de discrépance faible subit lui aussi cette malédiction : le nombre de points

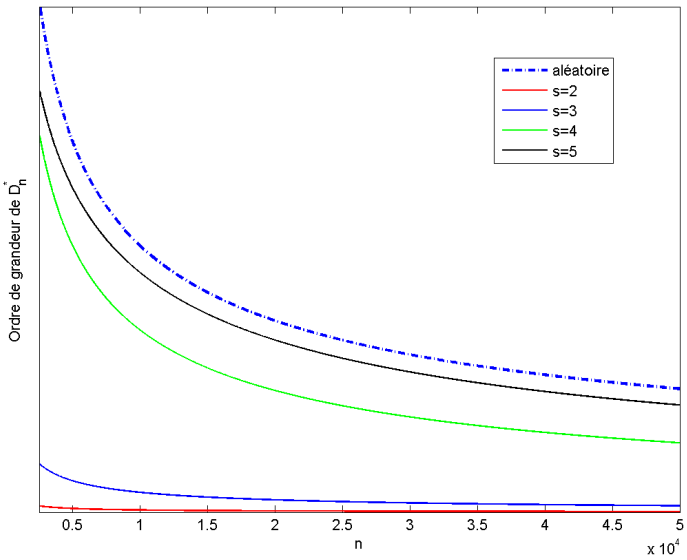



**FIGURE 2.9:** *Comparaison de l'évolution de l'ordre de grandeur de la discrépance star $D_n^*(x)$ d'une suite à faible dispersion par rapport à celle d'une suite aléatoire en fonction de la taille de la suite et de la dimension du problème : nous pouvons remarquer que lorsque la dimension augmente, le comportement de la discrépance des suites à discrépance faible se rapproche de celui d'une suite aléatoire.*

augmentent exponentiellement avec la dimension pour obtenir un certain niveau de discrépance. L'ajout d'aléatoire dans les suites à discrépance faible ne permet pas de contourner ce fléau de la dimension : il permet seulement d'améliorer la vitesses de génération des suites à faible discrépance. Lorsque la dimension augmente, les suites à discrépance faible se comportent de plus en plus comme des suites aléatoires. Les courbes présentées à la figure 2.9 confirment ce résultat. Lorsque la dimension augmente, le gain de l'utilisation des suites à faible discrépance devient de plus en plus faible par rapport au suites aléatoires : il s'agit là d'une limite de l'utilisation des suites à faible discrépance lorsque la dimension augmente.

## 2.4 Apprendre avec des suites à discrépance faible

Le but de cette partie est d'utiliser les suites à discrépance faible présentées à la section précédente dans le cadre de l'apprentissage actif. Nous rappelons que nous entendons par apprentissage actif, un système apprenant capable d'interroger un oracle sur l'étiquette de tout point de l'espace. De plus, nous supposons que ce système ne possède aucune connaissance sur le problème de régression à apprendre. Dans une première section, nous présentons les résultats théoriques existants, puis, dans une deuxième partie, nous présentons les résultats numériques en adéquation.

### 2.4.1 Résultats théoriques de l'apprentissage avec des suites à faible discrépance en régression

Les suites à discrépance faible ont été inventées pour faciliter l'estimation d'intégrales complexes et le théorème de Koksma-Hlawka (voir Niederreiter (1992)) permet alors de borner l'erreur d'estimation.

**Théorème de Koksma-Hlawka :** si $f$ est une application de $[0,1]^s$ dans $\mathbb{R}$, de variation au sens d'Hardy-Krause finie majorée par $V_{HK}(f)$, en utilisant la suite $\left(x_{(n)}\right)$ pour réaliser l'estimation, nous obtenons le résultat :

$$\left|\int f - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right| \leq V_{HK}(f)\, D_n^*\left(x_{(n)}\right).$$

La variation au sens d'Hardy-Krause est une mesure de régularité de la fonction $f$. Celle-ci est présentée, illustrée et commentée à l'annexe A à la page 125.
Ce théorème permet de borner l'erreur d'estimation en deux parties indépendantes : une partie dépendant uniquement de la régularité de la fonction intégrée, *i.e.* la variation d'Hardy-Krause, et une seconde partie dépendant uniquement de la régularité de l'échantillon utilisé pour l'estimation, *i.e.* sa discrépance. Ce théorème nous stipule donc qu'en minimisant la discrépance des suites, nous minimisons alors l'erreur d'estimation. Enfin, il est important de souligner que, contrairement aux méthodes de Monte Carlo, cette méthode de Quasi Monte Carlo, fournit une estimation déterministe de l'erreur et non une estimation stochastique.
Enfin, en utilisant des suites à discrépance faible, *i.e.* des suites dont la discrépance est inférieure à celle d'une suite aléatoire uniforme, on peut se rendre compte d'une amélioration de la borne de la vitesse de convergence de l'estimation. En, effet, elle passe de l'ordre de $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ à $\mathcal{O}\left(\frac{(\log(n))^s}{n}\right)$.

Remarquons qu'en considérant la discrépance comme la norme de la fonction écart entre la fonction de répartition empirique des données et la fonction de répartition de la loi uniforme, ce théorème d'Hardy-Krause s'écrit de la façon suivante :

$$\left|\int f - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right| = \left|\int_{\mathcal{X}} f(x)\, d\left(\hat{F}_n(x) - U(x)\right)\right| \leq V_{HK}(f)\, D_n^*\left(x_{(n)}\right).$$

**Théorèmes de Koksma-Hlawka sur des espaces de Hilbert à noyau reproduisant :** supposons que l'espace fonctionnel $\mathbb{H}$ soit un espace de Hilbert, dont toutes les fonctions définies sur $\mathcal{X}$ sont à variations finies au sens d'Hardy-Krause. Notons $K$ le noyau reproduisant de cet espace de Hilbert $\mathbb{H}$. Notons également $<.,.>_K$ le produit scalaire défini sur $\mathcal{X}\times\mathcal{X}$ par $K$, *i.e.* $<x,y>_K = K(x,y)$, et $\|.\|_K = \sqrt{<.,.>_K}$ la norme associée.
Par le théorème de représentation de Riesz (1955), il existe une fonction $\xi \in \mathbb{H}$ telle que :

$$\forall f \in \mathbb{H}, \left|\int_{\mathcal{X}} f(x) - \sum_{i=1}^{n} f(x_i)\right| = |<\xi, f>_K|.$$

La fonction $\xi$ s'écrit alors de la manière suivante :

$$\forall x \in \mathcal{X}, \xi(x) = \int_{\mathcal{X}} K(x,y)\, dy - \frac{1}{n}\sum_{i=1}^{n} K(x,x_i).$$

Par l'inégalité de Cauchy-Schwartz, nous pouvons montrer que :

$$\left|\int_{\mathcal{X}} f(x) - \sum_{i=1}^{n} f(x_i)\right| \leq \|f(.)\|_K \|\xi\|_K.$$

Ce résultat, démontré par Niederreiter et Spanier (1998), est connu sous le terme d'Inégalité de Koksma-Hlawka généralisée.
Nous pouvons interpréter $\|\xi\|_K$ comme la discrépance de P, et $\|f\|_K$ comme étant la variation au

sens de Hardy-Krause utilisée dans l'inégalité classique de Koksma-Hlawka.

Il est possible également de démontrer cette inégalité dans le cas des espaces de Banach, en remplaçant l'inégalité de Cauchy-Schwartz par l'inégalité de Hölder. L'inégalité de Hölder appliquée dans l'espace de Hilbert à paramètre $p = 1$ et $q = \infty$ et à noyau reproduisant correspond à l'inégalité classique de Koksma-Hlawka. Pour plus d'informations, on peut se référer à Hickernell (1998).

**Théorème de Koksma-Hlawka pour l'apprentissage :** l'utilisation des suites à discrépance faible a été initiée par Iwata et Ishii (2002) qui cherchaient à apprendre des formes en classification. Dans le cas de la régression active, Cervellera et Muselli (2004) sont les premiers à appliquer l'inégalité précédente. Ils obtiennent le résultat suivant :

- soit $f$ la fonction cible de $I^s$ dans $\mathbb{R}$ ;
- soit $l$ une fonction de coût ;
- soit $\mathbb{H}$ un espace fonctionnel dans lequel on cherche la fonction cible $f$. On suppose de plus que $\forall g \in \mathbb{H}, V_{HK}(l(f,g)) \leq M$ ;

alors pour tout $f \in \mathbb{H}$, on a :

$$\left|\mathscr{R}[f] - \hat{\mathscr{R}}_n[f]\right| \leq MD_n^*(x)$$

Et pour $\hat{f} \in \arg\min \hat{\mathscr{R}}[f]$, alors on a :

$$\hat{\mathscr{R}}[f] \leq \inf_{f\in\mathbb{H}} \mathscr{R}[f] + 2MD_n^*(x).$$

En utilisant une suite à discrépance faible, on peut alors obtenir la majoration suivante sous les mêmes conditions, *i.e.* $\hat{f} \in \arg\min \hat{\mathscr{R}}[f]$ :

$$\hat{\mathscr{R}}[f] \leq \inf_{f\in\mathbb{H}} \mathscr{R}[f] + 2Mc(s)\frac{(\log(n))^s}{n}.$$

où $c(s)$ est une constante dépendant uniquement de la dimension.

Autrement dit, ils prouvent ainsi la convergence de $\hat{\mathscr{R}}$ vers $\mathscr{R}^* = \arg\min_{g\in\mathbb{H}} \mathscr{R}[g]$ si pour toute fonction $g \in \mathbb{H}$, $f - g$ a une variation d'Hardy-Krause uniformément bornée.

Grâce à ce théorème, nous pouvons en déduire que l'utilisation des suites à faible discrépance améliore, comme pour l'estimation d'intégrales avec la méthode de (Quasi-) Monte Carlo, la borne de la vitesse de convergence en théorie. On peut remarquer aussi que les résultats obtenus sont déterministes et non stochastiques. Ces résultats restent cependant à nuancer. En effet, ils supposent que l'on soit capable d'estimer la discrépance de toute suite de données à utiliser. Enfin, cette approche suppose que la fonction de coût ait une variation au sens d'Hardy-Krause uniformément bornée par un réel $M$.

**Minimisation du risque structurel :** soit $f$ la fonction cible et prenons comme fonction de coût $L(f,g) = |f-g|$. On suppose maintenant que l'espace d'hypothèses $\mathbb{H}$ est tel que :

$$\forall g \in \mathbb{H}, V_{HK}(l(f,g)) \text{ est finie.}$$

Notons $\mathscr{R}^* = \inf_{f\in\mathbb{H}}$ et $f^* \in \arg\min \mathscr{R}$ ou alors $f^*$ est une fonction proche du minimum si celui-ci n'est pas atteint.

Supposons que $\hat{f}$ optimise $\hat{\mathscr{R}}_n[f] + c(s) V_{HK}(f) \frac{(\log(n))^s}{n}$ avec une précision $\varepsilon$. Alors Mary (2005) montre que :

$$\mathscr{R}\left[\hat{f}\right] \leq \mathscr{R}^* + \varepsilon + \left(V_{HK}(f^*) + V_{HK}(g) + V_{HK}(f^* - g)\right) c(s) \frac{(\log(n))^s}{n}.$$

Par rapport au corollaire précédent, à fonctions de coût équivalentes, ce résultat ne prend plus pour hypothèse le fait que la fonction de coût entre l'hypothèse cible $f$ et toute fonction de l'espace d'hypothèses est uniformément bornée sur cet espace d'hypothèses, *i.e.* que $\forall g \in \mathbb{H}, V_{HK}(l(f,g)) < M$. Cela est une hypothèse moins forte et permet de moins restreindre l'espace d'hypothèses.

**Borne sur l'erreur en généralisation :** supposons que l'estimation $\hat{f}$, obtenue par un algorithme d'apprentissage, converge simplement vers la fonction cible $f$ lorsque la taille de l'échantillon d'apprentissage $n \to +\infty$. En prenant pour fonction de coût l'erreur moyenne $l(f,g) = |f-g|$, alors Mary (2005) montre que :

$$\int |\hat{f} - g| \leq \frac{1}{n} \sum_{i=1}^{n} |\hat{f}(x_i) - f(x_i)| + \left(V_{HK}\left(\hat{f}\right) + V_{HK}(f)\right) D_n^*.$$

De plus, sous l'hypothèse que toutes les fonctions $\hat{f} - f$ sont bornées pour la norme de Hölder [2] de degré supérieur à $s$ non nécessairement entier, alors :

$$\int |\hat{f} - f| \leq \frac{1}{n} \sum_{i=1}^{n} |\hat{f}(x_i) - f(x_i)| + \left(2V_{HK}\left(\hat{f}\right) + \mathscr{O}(1)\right) D_n^*.$$

Ce résultat borne l'erreur d'estimation et se fonde uniquement sur la variation de la fonction cible ainsi que sur la discrépance de la suite d'apprentissage.

### 2.4.2 Expériences numériques d'apprentissage avec des suites à discrépance faible

L'utilisation des suites à discrépance faible en apprentissage a été initiée par Iwata et Ishii (2002) qui cherchaient à apprendre des formes en classification : nous reviendrons sur leurs expériences et sur le critère de la discrépance faible en classification active au chapitre 3.

Dans le cadre de l'apprentissage par régression, les premiers travaux théoriques sont dus à Cervellera et Muselli (2003b) et les premières expériences numériques sont dues à Cervellera et Muselli (2003a). A partir de ces travaux, ils appliquent la même méthodologie à différentes études. La première étude concerne l'apprentissage actif en présence d'un faible bruit d'étiquetage : Cervellera et Muselli (2004) montrent que la consistance de l'algorithme d'apprentissage est conservée et ils appliquent l'inégalité d'Hoeffding pour obtenir des vitesses de convergence du même ordre que celles obtenues en apprentissage statistique classique. Dans un deuxième temps, Cervellera et Muselli (2006) transforment un problème de contrôle en problème d'apprentissage actif et appliquent alors ces travaux. Enfin, dans une troisième étude, Cervellera *et al.* (2008) utilisent les résultats dans le cas d'apprentissage de fonctions de densité : la méthodologie reste inchangée, seules les expériences numériques varient.

Il est à noter que les expériences numériques présentées dans tous ces travaux (excepté pour Cervellera *et al.* (2008)) sont réalisées en dimension 4 et 6 sur les mêmes fonctions cibles. Ces fonctions sont :

$$g_1(x) = \sin(a.b) \text{ où } a = e^{2x_1 \sin(\pi x_4)} \text{ et } b = e^{2x_2 \sin(\pi x_3)}, \forall x \in [0,1]^4$$

2. La norme de Hölder de degré $b$ d'une fonction $f$ définie sur $\mathscr{X}$ est égale à : $\|f\|_b = \left(\int_{\mathscr{X}} \left|f(x)^b\right| dx\right)^{\frac{1}{b}}$.

$$g_2(x) = 4\left(x_1 - \frac{1}{2}\right)\left(x_4 - \frac{1}{2}\right)\sin\left(2\pi\sqrt{x_2^2 + x_3^2}\right), \forall x \in [0,1]^4$$

$$g_3(x) = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5 + x_6, \forall x \in [0,1]^6$$

L'apprentissage est réalisé avec un réseau de neurones à une couche cachée. Ils comparent les résultats obtenus sur des suites de tailles 500, 1000, 1500, 2000, 2500 et 3000. Ces suites sont des suites aléatoires et des suites de Niederreiter à faible discrépance. Les qualités d'apprentissage sont estimées à l'aide de la fonction de coût quadratique à partir de grilles régulières. Ces résultats soulignent l'intêret des suites à faible discrépance par rapport aux suites aléatoires uniformes. Enfin, il est à noter qu'aucune comparaison n'est réalisée entre apprentissage sur suites à discrépance faible et sur grilles régulières : en effet les grilles régulières ne sont pas à faible discrépance et l'intérêt de(s) l'étude(s) était de mettre en valeur l'effet de la faible discrépance, définie à partir de la discrépance d'une suite aléatoire uniforme.

Dans le cas de l'apprentissage de fonctions de densité, Cervellera *et al.* (2008) s'intéressent aux densités suivantes :

– la densité d'une loi log-normale en dimension 8, *i.e.*

$$f_1(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{(\ln(x)-\mu)^2/2\sigma^2}, \forall x \in [0,1\,000]^8$$

où $\mu \in \mathbb{R}$ et $\sigma > 0$ sont des paramètres à estimer ;

– la densité de la loi de Rayleight en dimension 10, *i.e.*

$$f_2(x) = \frac{x\exp\left(-\frac{x^2}{2\sigma^2}\right)}{\sigma^2}, \forall x \in [0,50]^{10}.$$

Comme précédemment, les auteurs utilisent un réseau de neurones avec une seule couche cachée et estiment la qualité d'apprentissage sur une grille avec un coût quadratique. Les apprentissages sont réalisés sur des suites aléatoires uniformes et sur des suites de Sobol. La taille des échantillons varie entre 1000, 2000, 3000 et 4000 points pour chaque loi. Ils tirent les mêmes conclusions que pour l' (les) expérience(s) précédente(s). Enfin, ils mettent en exergue également que les suites à faible discrépance ont une efficacité amoindrie lorsque la dimension augmente.

Enfin, dans sa thèse, Mary (2005) réalise expérimentalement un apprentissage actif de la fonction $f(x) = \sum_{i=1}^{s} x^i, \forall x \in I^s$. Les apprentissages sont réalisés sur des suites aléatoires uniformes et des suites à faible discrépance de Niederreiter avec des tailles allant jusqu'à 1 000 points en dimension $s = 7$ et jusqu'à 4 000 points en dimension $s = 14$. La fonction de risque est toujours l'erreur quadratique moyenne, et l'apprentissage est réalisé avec un SVR (Support Vector Regression) à noyau gaussien. Il observe lui aussi un gain significatif en utilisant les suites à faible discrépance par rapport aux suites aléatoires en dimension $s = 7$. Cependant, ce gain est beaucoup moins significatif en dimension 14. Les suites à discrépance faible sont donc aussi soumises à la malédiction de la dimension du problème. Ce constat est également réalisé dans les travaux de Teytaud *et al.* (2007) qui proposent alors d'utiliser des suites de Quasi Monte-Carlo ( *i.e.* des suites à faible discrépance) randomisées pour limiter la perte de puissance de ces suites en se basant sur les travaux de L'Ecuyer et Lemieux (2005). Un autre moyen pour essayer de repousser ce fléau est d'utiliser les méthodes de réduction de la variance dans l'estimation d'intégrales par Monte-Carlo.

**Méthodes de réduction de la variance dans l'estimation d'intégrales par Monte-Carlo :** en estimation d'intégrales par la méthode de Monte-Carlo ( *i.e.* avec des simulations aléatoires), le résultat obtenu est stochastique et possède donc une certaine variance. Afin de réduire celle-ci différentes méthodes ont été créées. Celles-ci, présentées non exhaustivement à l'annexe C, peuvent être appliquées au cas de l'apprentissage. La méthode de stratification est la plus adaptée à l'apprentissage de fonctions. En effet, dans de nombreux problèmes d'apprentissage, les données sont stratifiées : par exemple elles arrivent en ligne en fonction de connections à un site internet ou alors elles sont issues de sondages réalisés par des instituts de sondages qui, pour une étude, préfèrent qu'une certaine catégorie soit sur-représentée. Un « clustering » des données peut également être réalisé avant apprentissage (voir Devroye *et al.*, 1997, Chap. 21) : il est alors possible d'apprendre localement sur chaque cluster, définissant ainsi le modèle d'apprentissage par une juxtaposition de modèles locaux (la méthodologie est expliquée à la section 6.4 dans le cadre de la sélection de points). Cependant, ce type d'apprentissage n'entre pas dans le contexte dans lequel nous nous sommes positionnés. Enfin, ces méthodes ne permettent pas de repousser significativement ce problème de vitesse de convergence des suites à faible discrépance.

## 2.5 Conclusion

Dans ce chapitre, nous abordons le problème de la génération des premiers points d'apprentissage dans le cas de la régression ou de l'apprentissage de nappes. L'uniformité des points d'apprentissage est identifiée comme étant le critère important. Nous avons alors présenté une forme d'uniformité : la discrépance, et avons alors défini le concept de suites à discrépance faible.

L'utilisation des suites à discrépance faible est bien connue en ingénierie pour la construction de méta-modèles. Par exemple, dans sa thèse réalisée au CEA [3], Feuillard (2007) utilise le critère de discrépance pour qualifier la qualité des bases de données, ainsi que pour choisir la meilleure base parmi un ensemble de bases afin de construire un méta-modèle du phénomène étudié. Dans une autre optique, il utilise aussi le (les) critère(s) de la discrépance pour sélectionner les points à utiliser dans une riche base de données pour construire un méta-modèle. Un autre exemple est la thèse de Franco (2008) qui, toujours dans le cas de la construction de méta-modèles en ingénierie, utilise les suites à discrépance faible car elles ont de bonnes propriétés de remplissage de l'espace pour capter les non linéarités des phénomènes, ainsi qu'une bonne répartition des points en projection dans le cas où le phénomène ne dépendrait que de quelques variables influentes.

Dans ce chapitre, nous avons utilisé ces suites à discrépance faible et avons mis en évidence l'avantage de l'utilisation des suites à discrépance faible par rapport à la distribution uniforme stochastique : vitesse de convergence plus rapide et convergence déterministe sous l'hypothèse de variation bornée des fonctions. Nous avons également présenté différents résultats numériques réalisés par différents auteurs : les résultats confirment la théorie, mais se heurtent cependant à la malédiction de la dimension.

Dans le prochain chapitre, nous essayons de transférer ces résultats au cas de la classification ou apprentissage de variétés.

3. Commissariat à l'Énergie Atomique et aux Énergies Alternatives.

# - CHAPITRE 3 -

# LES RÉSULTATS THÉORIQUES SUR LE PROBLÈME DE RÉGRESSION NE SE TRANSFÈRENT PAS AU PROBLÈME DE CLASSIFICATION

## Sommaire



DANS CE CHAPITRE, nous nous intéressons à la génération des premiers points d'apprentissage dans le contexte des problèmes de classification ou d'apprentissage de variétés.

Idéalement, de façon similaire à l'apprentissage de nappes au chapitre 2, les zones de l'espace où la fonction varie beaucoup doivent bénéficier d'un échantillonnage plus important, alors que les zones où la fonction varie peu ou pas ne doivent bénéficier que d'un échantillonnage plus superficiel ou nul. En apprentissage de variétés (ou classification) cela signifie que l'échantillonnage doit être intensif aux endroits de changement de classe. Une représentation graphique de ces zones devant bénéficier d'un fort échantillonnage dans un cadre de la classification est donnée à la figure 3.1.

Lors de ce premier échantillonnage des points d'apprentissage, la forme de la variété est cependant inconnue. Afin d'obtenir en moyenne la meilleure approximation, les points doivent alors être générés « uniformément ». Nous avons vu que, dans le cas d'apprentissage de fonctions, *i.e.* régression, l'« uniformité » selon la discrépance est le critère adéquat.

En apprentissage, certaines écoles considèrent que la classification est un cas particulier de la régression : il s'agit du cas où la fonction cible est continue par morceaux, et est donc une fonction qui évolue peu. Par conséquent, les méthodes de génération des premiers points « uniformes » d'ap-

prentissage en régression doivent s'adapter au cas de la classification.

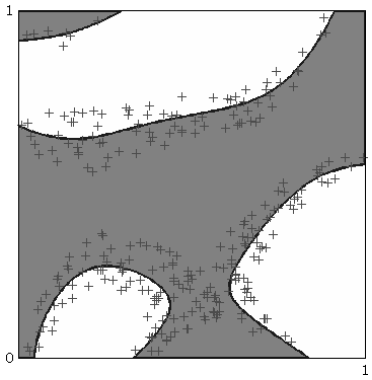



**FIGURE 3.1:** *Exemple d'échantillonnage adapté pour la classification (ou l'apprentissage d'une variété) en dimension 2 : les points sont plus nombreux dans les zones où la fonction varie beaucoup, i.e. à la frontière, dans les zones de changement de classes. Mais au départ cette frontière est inconnue, en général.*

Iwata et Ishii (2002) ont montré que « dérandomiser [1] » l'échantillon d'apprentissage permettait d'obtenir de meilleurs résultats en classification : ils proposent également d'utiliser le critère de faible discrépance pour générer cet échantillon par rapport à un échantillon aléatoire. Remarquons que dans leur approche, ils ne se soucient pas de la variation d'Hardy-Krause présente dans le théorème de Koksma-Hlawka.

Dans ce chapitre, nous nous intéressons donc en détail au transfert à la classification du critère de discrépance de la régression. Pour cela, nous considérons la classification comme un cas particulier de régression. Puis nous essayons de transférer théoriquement les résultats sur la discrépance en classification en comparant des apprentissages réalisés sur grilles régulières et sur des suites à faible discrépance. Nous concluons que ces résultats ne sont pas applicables au problème de classification.

Ce chapitre a fait l'objet de communications à une conférence francophone (Gandar *et al.* (2009a)), à une conférence européenne (Gandar *et al.* (2009b)) et fait l'objet actuellement d'une soumission à une revue scientifique (Gandar *et al.* (subm)).

## 3.1 Positionnement de la classification par rapport à la régression

Dans cette section, nous présentons différents points de vue sur la classification par rapport à la régression. Notre but n'est pas de nous positionner par rapport à une définition, mais d'essayer de comprendre comment un problème de classification peut être vu comme un problème de régression.

**La classification est incluse dans la régression :** au début de cette thèse, nous avons défini la régression, ou l'apprentissage de fonctions, comme le cas où la fonction cible $f$ est une fonction de $\mathbb{R}^s$ dans $\mathbb{R}$. Nous avons défini la classification comme le cas où la fonction cible $f$ est une fonction indicatrice et prend ses valeurs dans un ensemble de cardinalité finie, le plus souvent l'ensemble $\{0,1\}$. Cet ensemble étant inclus dans $\mathbb{R}$, nous pouvons donc considérer la classification comme un cas particulier de régression. Cette caractérisation de la régression et de la classification est réalisée

1. dérandomiser : verbe transitif du 1er groupe (anglais *random*), action d'enlever un caractère aléatoire à un processus. Ce verbe n'existe pas dans le dictionnaire Français. Espérons qu'il y apparaisse rapidement ! ;-)

en se basant sur l'espace d'arrivée de la fonction cible et non sur la continuité de celle-ci. Ce point de vue est utilisé intrinsèquement par Iwata et Ishii (2002) qui développent la méthodologie d'apprentissage actif pour la classification sans se soucier de la continuité de la fonction cible et/ou des fonctions de coût.

**La classification diffère de la régression :** en considérant la cardinalité de l'espace d'arrivée de la fonction cible, ainsi que la continuité de cette fonction, Vapnik et Chervonenkis (1971) différencient autrement la régression de la classification : i) si l'espace d'arrivée est dense, et que la fonction cible est continue, il s'agit de la régression et ii) si l'espace d'arrivée est de cardinalité finie, il s'agit de la classification. Le cas où l'espace de la fonction cible est dense, mais que cette dernière n'est pas continue, est un cas mixte des définitions précédentes et reste un cas exotique : nous ne le considérons pas dans notre science.

**La classification : un cas particulier de la régression ?** Castro *et al.* (2005) considèrent que la classification est un cas particulier de la régression dans le sens où la fonction cible est partout constante, sauf au niveau de la frontière : ils appellent cette classe de fonctions la classe des fonctions constantes par morceaux (en anglais, *Piecewise Constant*) notée $PC(\beta, M)$. Afin de traduite mathématiquement ces fonctions, commençons par déterminer une fonction localement constante : une fonction $f : [0,1]^s \to \mathbb{R}$ est dite localement constante en un point $x \in: [0,1]^s$ si :

$$\exists \varepsilon > 0 : \forall y \in [0,1]^s : \|x - y\| < \varepsilon \Rightarrow f(y) = f(x).$$

Une fonction $f$ est dite « Piecewise Constant » si

- $f$ est uniformément bornée, *i.e.* $\exists M > 0, \forall x \in [0,1]^s, |f(x)| \leq M$ ;
- $f$ est localement constante en tout point $x \in [0,1]^s$ $|B(f)$ où $B(f)$ est un ensemble de points contenus dans une boîte de dimension au plus égale à $s-1$ ;
- $B(f)$ satisfait $N(r) \geq \beta r^{-(s-1)}$ pour tout $r > 0$, où $\beta > 0$ est une constante et $N(r)$ est le nombre minimal de boules fermées de diamètre $r$ couvrant $B(f)$.

Avec cette définition, une surface de classification, ou variété, est vue comme étant une fonction constante par morceaux, et donc une fonction de régression. La dernière condition permet d'avoir des fonctions relativement lisses, empêchant ainsi d'avoir des frontières trop compliquées ; les paramètres $\beta$ et $N(r)$ permettant de catégoriser les fonctions. Typiquement cela empêche en cas extrême d'être confronté à des frontières de classification fractales, comme comme l'exemple présenté à titre illustratif à la figure 3.2.

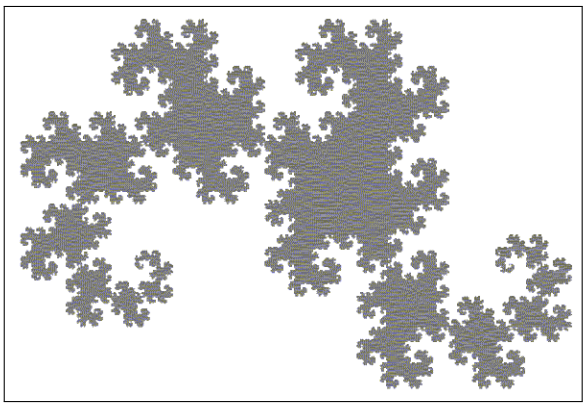


**FIGURE 3.2:** *Exemple de variétés fractales : la courbe du Dragon. La frontière de classification se répète à l'infini. Image provenant de Monnerot-Dumaine (2012).*

Comme nous l'avons constaté, la classification peut être vue comme un cas particulier de la régression en omettant la notion de continuité de la fonction (nonobstant l'importance de cette dernière), et/ou en considérant la fonction cible comme constante par morceaux. A ce titre, les résultats issus de l'apprentissage actif en régression peuvent être transférés à la classification. Cette omission de la continuité, aussi bien de la fonction cible, que des fonctions de coût n'est cependant pas anodine. Les premiers « Machine Learner » qui s'intéressèrent à la problématique de génération déterministe des premiers points d'apprentissage en classification, *i.e.* Iwata et Ishii (2002), ont fait cette omission.

Dans la prochaine section, nous allons étudier le transfert théorique des résultats de l'apprentissage actif en régression à l'apprentissage actif en classification. Dans un premier temps, nous nous abstenons d'abord de cette notion de continuité pour réaliser l'étude théorique. Dans un deuxième temps, nous prenons en compte cette notion de continuité et voyons que le transfert de ces résultats ne peut être réalisé que sous une hypothèse très restrictive qui n'est jamais vérifiée en pratique. Puis, dans la section suivante, nous nous intéressons à l'utilisation des suites à discrépance faible en pratique. Dans une troisième et dernière section, nous faisons une conclusion sur l'ensemble de ces travaux.

## 3.2 Étude théorique de la classification active avec la discrépance

Dans la section précédente, nous avons discuté de la continuité des fonctions. Dans cette section, nous nous intéressons au transfert de ces résultats, de la régression active à la classification active. Nous commençons cette section en ne prenant pas en compte la (non-) continuité des fonctions cibles et de coût. Puis, nous nous interrogeons sur le rôle que celle-ci joue dans les résultats.

### 3.2.1 Apprentissage statistique *vs* apprentissage actif sur la discrépance

Nous savons que la classification correspond au cas où la fonction cible $f$ est une fonction indicatrice et prend ses valeurs dans un ensemble discret, du type $\{0,1\}$. En utilisant le théorème de Koksma-Hlawka présenté à la section 2.4.1, nous obtenons le même résultat que pour l'apprentissage en régression, *i.e.* que pour tout $g \in \mathbb{H}$ :

$$\left|\mathscr{R}[f] - \hat{\mathscr{R}}_n[f]\right| \leq V_{HK}(l(f,g))\, D_n^*\left(x_{(n)}\right) \tag{3.1}$$

où $V_{HK}(l(f,g))$ est la variation d'Hardy-Krause de la fonction de coût utilisée.

Dans cette partie, nous faisons l'hypothèse forte que l'algorithme d'apprentissage est tel que la fonction de coût utilisée, à deux modalités, ne possède que des discontinuités selon les axes de direction de l'espace. La figure 3.3 représente un exemple de fonction de coût possédant une telle propriété. Autrement dit, les zones de l'espace dans lesquelles on apprend mal, sont composées de la juxtaposition finie de rectangles. En réalisant cette hypothèse, nous nous situons ainsi dans une bonne condition mathématique : la variation d'Hardy-Krause de la fonction de coût, *i.e.* le terme $V_{HK}(l(f,g))$ dans l'équation 3.1, est alors finie numériquement. Ceci implique donc que l'erreur d'approximation est utilisable en la bornant de manière finie. Nous revenons sur cette hypothèse et ce qu'elle implique dans le paragraphe 3.2.2.

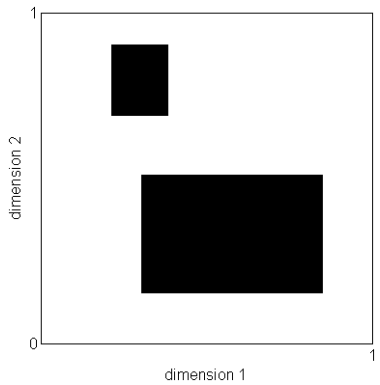



**FIGURE 3.3:** *Exemple de graphe d'une fonction indicatrice à variation d'Hardy-Krause finie en dimension 2. Les seules discontinuités de la fonction sont parallèles aux axes de direction de l'espace. Autrement dit, une des classes de la fonction (la noire sur la figure) est composée d'une juxtaposition finie d'hyper-rectangles.*

**Comparaison à la théorie d'apprentissage selon Vapnik (1995) :** notons $f^* \in \arg\min_{f \in \mathbb{H}} \mathscr{R}[f]$ la fonction minimisant le risque réel et $\hat{f}$ une estimation de la fonction cible $f$. Vapnik (1995) montre que sous l'hypothèse d'une dimension de Vapnik-Chervonenkis finie de l'espace d'hypothèse $\mathbb{H}$, et si $\hat{\mathscr{R}}_n[f^*] = 0$, alors $\hat{\mathscr{R}}_n\left[\hat{f}\right]$ décroît à une vitesse en $\frac{1}{n}$ avec un niveau de confiance de $1-\delta$.

En utilisant la théorie de l'apprentissage avec la discrépance, nous obtenons les différences suivantes :

- l'hypothèse de dimension de Vapnik-Chervonenkis finie est remplacée par l'hypothèse de variation d'Hardy-Krause finie (nous reviendrons sur cette hypothèse au paragraphe 3.2.2) ;
- la borne d'erreur dépend de la fonction hypothèse (de par sa variation) ;
- vitesse de convergence déterministe mais présence d'un terme supplémentaire en $(\log(n))^s$ ;
- il n'est pas nécessaire d'avoir l'hypothèse $\hat{\mathscr{R}}_n[f^*] = 0$ comme dans la théorie de Vapnik-Chervonenkis pour obtenir une vitesse de convergence en $\frac{1}{n}$ au lieu de $\frac{1}{\sqrt{n}}$.

**Comparaison à la minimisation du risque empirique :** dans la théorie de minimisation du risque empirique, Vapnik (1995) considère une suite emboitée de famille d'hypothèses à dimension de Vapnik-Chervonenkis finie. Ils montrent alors que $\mathscr{R}[g]$ décroît en $\sqrt{\frac{\log(n)}{n}}$ avec un niveau de confiance $1-\delta$.
En utilisant la théorie de l'apprentissage avec la discrépance, nous obtenons les différences suivantes :

- vitesse de convergence en $\frac{\log^s(n)}{n}$ au lieu de $\sqrt{\frac{\log(n)}{n}}$ ;
- vitesse de convergence déterministe.

Remarquons enfin que la dimension $s$ n'intervient pas visuellement dans la vitesse de convergence de l'apprentissage statistique. Néanmoins, elle est contenue dans la dimension de Vapnik-Chervonenkis de l'espace fonctionnel.

### 3.2.2 Borne d'erreur théorique en classification active avec la discrépance

Dans la section précédente, nous nous sommes placés dans un contexte particulier d'apprentissage dans lequel les variations d'Hardy-Krause des fonctions coûts sont finies. Nous nous situons maintenant dans le cas général des fonctions de coût : celles-ci sont de forme indicatrice quelconque. Avec le théorème de Koksma-Hlawka, nous obtenons le même résultat, *i.e.* que pour tout $g \in \mathbb{H} : \left|\mathscr{R}[f] - \hat{\mathscr{R}}_n[f]\right| \leq V_{HK}(l(f,g)) D_n^* \left(x_{(n)}\right)$.

Cependant, la variation d'Hardy-Krause de fonctions indicatrices est généralement infinie : Owen (2004) montre que seules les fonctions indicatrices dont les discontinuités sont parallèles aux axes possèdent une variation d'Hardy-Krause finie. Ainsi, le théorème d'Koksma-Hlawka pour l'apprentissage de variétés devient alors :

$$\left|\mathscr{R}[f] - \hat{\mathscr{R}}_n[f]\right| \leq +\infty.$$

Par conséquent, sous cette hypothèse de fonction de coût indicatrice quelconque, la théorie de l'apprentissage en régression, fondée sur l'utilisation des suites à faible discrépance et sur le théorème de Koksma-Hlawka, ne peut être transférée théoriquement à la classification. Sous l'hypothèse inverse de fonction de coût indicatrice non quelconque à discontinuités parallèles aux axes, *i.e.* que l'apprentissage est réalisé de telle sorte que les erreurs d'apprentissage se contentent d'être dans des rectangles parallèles aux axes, la théorie d'apprentissage basée sur la discrépance se transfère sans problèmes. Cependant, cette hypothèse d'apprentissage est une hypothèse très forte et ne peut jamais être garantie en pratique.

Ce résultat nous stipule donc qu'en général, la théorie sur l'utilisation des suites à discrépance faible en régression ne se transfère pas au cas de la classification. Il ne nous stipule en rien que les suites à faible discrépance ne sont pas adaptées à l'apprentissage de variétés. Nous démontrons ce résultat expérimentalement à la section suivante.

## 3.3 Des exemples confirment que la classification n'a pas le même comportement que la régression vis à vis de la discrépance

Dans la section précédente, nous avons mis en évidence que la théorie sur l'utilisation des suites à discrépance faible adaptée à la classification n'est pas valide. Dans cette section, nous nous intéressons à l'utilisation expérimentale de cette théorie. Nous suivons alors le comportement de la qualité d'apprentissage en fonction de la discrépance de l'échantillon d'apprentissage afin d'affirmer ou de confirmer cette validité théorique.

Dans cette optique, nous générons alors un ensemble artificiel de règles de classification comme présenté à l'annexe B, page 129. Pour chaque règle, nous apprenons avec trois échantillons différents de même taille et trois méthodes d'apprentissage. Le premier échantillon est une grille régulière [2] et les deux autres sont des suites de Halton et de Sobol définies aux chapitres 2.2.2 et 2.2.5. Ces suites de Halton et de Sobol sont deux suites différentes, construites à partir d'algorithmes différents, et qui possèdent une faible discrépance de l'ordre de $\mathscr{O}\left(\frac{\log^s(n)}{n}\right)$.

Les algorithmes d'apprentissage que nous avons utilisés sont les $k$-NN (ou $k$ plus proches voisins), l'algorithme des SVMs (Support Vector Machine ou Séparateurs à Vaste Marge, voir Shawe-Taylor et Cristianini (2000); Cornuéjols (2002)) et les arbres de classification avec ces configurations :

2. Grille régulière disposant de points sur les arêtes de l'hyper-cube.

- $k$-NN : validation croisée sur 5 échantillons basée sur l'erreur d'apprentissage (pour chaque règle et chaque échantillonnage) pour choisir la taille de voisinage dans l'ensemble $\{3,5,7\}$.
- SVM : validation croisée sur 5 échantillons basée sur l'erreur d'apprentissage (pour chaque règle et chaque échantillonnage) pour choisir la largeur de bande $\sigma$ du noyau gaussien entre les valeurs 0.2 et 2. Le paramètre $C$ est fixé [3] à 10 000, et nous avons travaillé avec la boîte à outils libSVM développée par Chang et Lin (2001).
- arbre : nous divisons les nœuds de l'arbre qui possèdent 3 individus ou plus, et nous travaillons avec l'algorithme décrit par Breiman (1993).

Les tables 3.1 et 3.2 montrent les erreurs en généralisation sur 1 000 règles en dimension 2 et 3 pour une table, et en dimension 4 et 5 pour l'autre. Ces erreurs sont estimées sur des séquences régulières de 6 000 points en utilisant la fonction de coût classique.

Pour chaque ensemble de problèmes et chaque méthode d'apprentissage, nous avons testé statistiquement la pertinence de ces résultats : nous avons réalisé un test statistique de Student au niveau de signification de 1%. Nous avons attribué le rang 1 à l'échantillonnage ayant la moyenne d'erreur en généralisation la plus basse. De façon identique, nous avons alloué le rang 3 à l'échantillonnage ayant la moyenne d'erreur la plus haute.

Nous pouvons remarquer qu'en général la moyenne d'erreur en généralisation sur grille est inférieure à celle obtenue sur les suites à discrépance faible. Il est clair expérimentalement que la faible discrépance ne permet pas d'obtenir les meilleurs résultats en apprentissage de variétés car nous obtenons de meilleurs résultats sur les grilles, qui ne sont pas à discrépance faible. Par conséquent, la classification ne possède pas le même comportement que la régression vis à vis de la discrépance. Ainsi, minimiser la discrépance ne semble pas être le critère adéquat pour générer l'échantillonnage optimal d'un problème de classification.

Enfin, Morokoff et Caflisch (1995) et Press et Teukolsky (1989) ont démontré expérimentalement en estimation d'intégrales, qu'utiliser la faible discrépance n'est pas efficace lorsque les intégrandes sont des fonctions non lisses ou possèdent des discontinuités : ceci est typiquement le cas des fonctions indicatrices de classification. Ce résultat apporte un argument supplémentaire à l'hypothèse selon laquelle la discrépance faible n'est pas le critère adéquat pour générer des points d'apprentissage en classification.

## 3.4 Conclusion

Dans ce chapitre,nous nous sommes intéressés au transfert des résultats d'apprentissage en régression active au cas de la classification active. Nous nous sommes tout d'abord interrogés sur la place de la classification dans la théorie de l'apprentissage de fonctions. Nous avons pu, sans prendre parti, voir différents points de vue selon lesquels la classification peut être à la fois considérée comme cas particulier de la régression et comme un cas distinct.

Nous distinguons la classification de la régression, principalement par la non-continuité de la fonction cible, mais également, et c'est lié, par la non-continuité des fonctions de coût associées. Cette différentiation est importante dans l'étude théorique de la classification active. En effet, si nous ne prenons pas en compte cette non-continuité, la classification se comporte alors comme la régression. Néanmoins, cette hypothèse de non-continuité pour laquelle la théorie s'applique correspond à un type d'apprentissage très particulier [4], qui a une probabilité quasi-nulle de se rencontrer.

3. Nos problèmes sont « hard-margin ».
4. Les erreurs d'apprentissage se localisent dans une juxtaposition finie de rectangles de côtés parallèles aux axes.

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Grille | | Suite de Halton | | Suite de Sobol | |
|---|---|---|---|---|---|---|---|---|
| | | | valeur | *rang* | valeur | *rang* | valeur | *rang* |
| 2 | 100 | SVM | **3,20** | *1* | 3,43 | *2* | 3,70 | *3* |
| | | kNN | **3,32** | *1* | 4,21 | *2* | 4,62 | *3* |
| | | Arbre | **4,62** | *1* | 6,05 | *2* | 6,26 | *3* |
| | 225 | SVM | **1,98** | *1* | 2,13 | *2* | 2,14 | *2* |
| | | kNN | **2,12** | *1* | 2,82 | *2* | 2,72 | *2* |
| | | Arbre | **3,03** | *1* | 4,09 | *2* | 4,19 | *2* |
| | 400 | SVM | **1,37** | *1* | 1,53 | *2* | 1,55 | *2* |
| | | kNN | **1,64** | *1* | 2,14 | *2* | 2,22 | *2* |
| | | Arbre | **2,27** | *1* | 3,17 | *2* | 3,35 | *3* |
| | 676 | SVM | **1,03** | *1* | 1,13 | *2* | 1,13 | *2* |
| | | kNN | **1,18** | *1* | 1,68 | *2* | 1,67 | *2* |
| | | Arbre | **1,74** | *1* | 2,50 | *2* | 2,56 | *2* |
| | 2 500 | SVM | **0,48** | *1* | 0,51 | *2* | 0,54 | *2* |
| | | kNN | **0,59** | *1* | 0,82 | *2* | 0,98 | *3* |
| | | Arbre | **0,87** | *1* | 1,36 | *2* | 1,41 | *2* |
| 3 | 64 | SVM | **7,84** | *1* | 8,59 | *2* | 8,23 | *2* |
| | | kNN | **8,38** | *1* | 9,84 | *3* | 9,48 | *2* |
| | | Arbre | **12,07** | *1* | 13,90 | *2* | 13,53 | *2* |
| | 512 | SVM | **3,10** | *1* | 3,46 | *2* | 3,37 | *2* |
| | | Arbre | **5,14** | *1* | 7,34 | *2* | 7,33 | *2* |
| | 1 000 | SVM | **2,23** | *1* | 2,50 | *2* | 2,50 | *2* |
| | | kNN | **2,75** | *1* | 3,64 | *2* | 3,64 | *2* |
| | | Arbre | **4,08** | *1* | 5,88 | *2* | 5,81 | *2* |
| | 3 375 | SVM | **1,27** | *1* | 1,43 | *2* | 1,42 | *2* |
| | | kNN | **1,76** | *1* | 2,33 | *2* | 2,29 | *2* |
| | | Arbre | **2,71** | *1* | 3,96 | *2* | 3,87 | *2* |

**TABLE 3.1:** *Moyenne et rang de l'erreur en généralisation sur 1 000 règles de classification en dimension 2 et 3 (en %). Pour chaque ensemble de règles et chaque méthode d'apprentissage, les meilleurs résultats sont inscrits en gras. Lorsque le test de Student ne détecte pas une différence significative, les rangs sont identiques.*

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Grille | | Suite de Halton | | Suite de Sobol | |
|---|---|---|---|---|---|---|---|---|
| | | | valeur | *rang* | valeur | *rang* | valeur | *rang* |
| 4 | 625 | SVM | **5,26** | *1* | **5,22** | *1* | **5,24** | *1* |
| | | kNN | **6,57** | *1* | 6,77 | *2* | 6,83 | *2* |
| | | Arbre | **9,60** | *1* | 11,18 | *2* | 10,91 | *2* |
| | 1 296 | SVM | **3,85** | *1* | 3,98 | *2* | 4,00 | *2* |
| | | kNN | **4,93** | *1* | 5,57 | *2* | 5,69 | *2* |
| | | Arbre | **7,08** | *1* | 9,50 | *2* | 9,54 | *2* |
| | 2 401 | SVM | **2,92** | *1* | 3,01 | *2* | 3,15 | *2* |
| | | kNN | **4,16** | *1* | 4,78 | *2* | 4,82 | *2* |
| | | Arbre | **6,01** | *1* | 8,26 | *2* | 8,27 | *2* |
| | 4 096 | SVM | **2,50** | *1* | **2,53** | *1* | **2,58** | *1* |
| | | kNN | **4,21** | *1* | **4,23** | *1* | **4,29** | *1* |
| | | Arbre | **6,10** | *1* | 7,29 | *2* | 7,23 | *2* |
| | 6 561 | SVM | **1,44** | *1* | 2,12 | *2* | 2,13 | *2* |
| | | kNN | **2,14** | *1* | 3,78 | *2* | 3,72 | *2* |
| | | Arbre | **2,25** | *1* | 6,53 | *2* | 6,52 | *2* |
| | 10 000 | SVM | **1,65** | *1* | 1,81 | *2* | 1,81 | *2* |
| | | kNN | **2,61** | *1* | 3,37 | *2* | 3,34 | *2* |
| | | Arbre | **3,78** | *1* | 6,02 | *2* | 6,02 | *2* |
| 5 | 243 | SVM | **10,30** | *1* | 10,95 | *2* | **10,26** | *1* |
| | | kNN | **11,27** | *1* | 11,80 | *2* | **11,15** | *1* |
| | | Arbre | **13,43** | *1* | 18,54 | *2* | 18,14 | *2* |
| | 1 024 | SVM | **5,61** | *1* | 6,51 | *2* | 6,35 | *2* |
| | | kNN | **7,83** | *1* | 8,29 | *2* | 7,98 | *2* |
| | | Arbre | **10,43** | *1* | 14,60 | *2* | 14,23 | *2* |
| | 3 125 | SVM | **4,40** | *1* | 4,54 | *2* | 4,54 | *2* |
| | | kNN | **4,96** | *1* | 5,74 | *2* | 5,77 | *3* |
| | | Arbre | **9,60** | *1* | 12,10 | *2* | 11,87 | *2* |
| | 7 776 | SVM | **2,78** | *1* | 4,00 | *2* | 3,36 | *2* |
| | | kNN | **3,48** | *1* | 5,40 | *2* | 5,36 | *2* |
| | | Arbre | **4,37** | *1* | 10,30 | *2* | 10,36 | *2* |
| | 16 807 | SVM | **2,31** | *1* | 2,66 | *2* | 2,63 | *2* |
| | | kNN | **3,83** | *1* | 4,64 | *2* | 5,57 | *2* |
| | | Arbre | **5,46** | *1* | 8,97 | *2* | 8,98 | *2* |

**TABLE 3.2:** *Moyenne et rang de l'erreur en généralisation sur 1 000 règles de classification en dimension 4 et 5 (en %). Pour chaque ensemble de règles et chaque méthode d'apprentissage, les meilleurs résultats sont inscrits en gras. Lorsque le test de Student ne détecte pas une différence significative, les rangs sont identiques.*

En prenant en compte cette non-continuité, nous ne pouvons transférer les résultats théoriques. Ces résultats nous permettent alors d'avancer un début de réponse à la question « la classification, un cas particulier de la régression ? » : la classification ne semble pas être un cas particulier de la régression.

En pratique, nous nous sommes intéressés à l'utilisation des suites à discrépance faible en classification. De manière surprenante, nous avons pu constater que nous obtenons de meilleurs résultats sur les grilles que sur les suites à faible discrépance. Or les grilles ne sont pas à faible discrépance. Ces résultats ont été obtenus avec différentes règles de classification, avec différentes dimensions et avec trois algorithmes d'apprentissage différents afin de s'affranchir de l'algorithme d'apprentissage. Il semble donc que le comportement de la classification soit différent de celui de la régression par rapport à la discrépance. Il apparaît donc que la discrépance ne soit pas le critère adéquat pour générer les premiers points d'apprentissage en classification active.

A la vue de ces résultats, nous pouvons donc dire que les résultats de la régression ne se transfèrent pas théoriquement et expérimentalement à la classification. Nous pouvons également délivrer une réponse ferme à la question fil d'Ariane de ce chapitre : non, du point de vue de la génération des premiers points d'apprentissage, la classification n'est pas un cas particulier de la régression !

Iwata et Ishii (2002) ont montré que dérandomiser l'échantillon d'apprentissage en utilisant des suites à faible discrépance permettait d'obtenir de meilleurs résultats. Nous ne contestons nullement ce résultat. Avec nos expériences numériques, nous avons montré que les grilles régulières permettent d' obtenir de meilleurs résultats. Nous pouvons alors hiérarchiser les plans d'expériences en fonction de leur « efficacité croissante » en apprentissage de variétés de la manière suivante : plans aléatoires $<$ suites à faible discrépance $<$ grilles régulières. Autrement dit, nous avons mis en évidence que la discrépance faible n'est pas le meilleur critère de génération de points pour la classification.

Dans la suite de la thèse, nous nous intéresserons donc à l'explication de ce résultat. Celui-ci amène alors naturellement les questions suivantes : pourquoi la discrépance n'est-elle pas appropriée au cas de la classification ? Qu'est ce qui, à part la discrépance, caractérise les grilles régulières par rapport aux suites à faible discrépance ?

# - CHAPITRE 4 -

# INITIALISATION DANS LE CAS DE LA CLASSIFICATION : LA DISPERSION, UN NOUVEAU CRITÈRE ?

## Sommaire



DANS LE CHAPITRE PRÉCÉDENT, nous avons mis en évidence que la discrépance n'est pas le critère adéquat pour générer les premiers points d'apprentissage. Nous nous posions alors les questions suivantes : pourquoi la discrépance n'est-elle pas appropriée au cas de la classification ? Qu'est-ce qui, autre que la discrépance, caractérise les grilles régulières par rapport aux suites à faible discrépance ?
Dans ce chapitre, nous essayons de trouver une caractérisation, différente de celle par la discrépance, qui permettrait d'expliquer cette différence de qualité d'apprentissage.

En s'inspirant des premiers travaux d'optimisation de fonctions de Niederreiter (1988), nous présentons dans une première partie le critère de la dispersion comme nouveau critère d'« uniformité » d'une suite. Dans une deuxième partie, nous revenons sur les résultats numériques présentés à la section 3.3 en se focalisant sur la dispersion des échantillons d'apprentissage : nous conjoncturons et proposons alors le critère de dispersion comme un critère adéquat pour générer des points d'apprentissage en classification. Nous montrons dans une troisième partie, que le critère de dispersion pour générer les premiers points d'apprentissage en classification est un critère adéquat. Cette démonstration est réalisée théoriquement en établissant un lien théorique entre qualité d'apprentissage et dispersion dans un contexte particulier de classification. Nous proposons alors une caractérisation particulière de la régularité des variétés. Enfin, nous élargissons cette démonstration

empiriquement sur un ensemble conséquent d'expériences.

Ce chapitre a fait l'objet, en plus des communications présentées au chapitre 3, d'une présentation à une conférence internationale (Gandar *et al.* (2011)).

## 4.1 La dispersion d'une suite

En analyse numérique, la recherche d'un optimal global d'une fonction est une question d'optimisation fréquente. Dans le cas où la fonction étudiée possède un très faible degré de régularité, *e.g.* dans le cas de fonction non différentiable, les méthodes classiques utilisant la descente de gradient ne sont pas employées et il est alors d'usage de se tourner vers des méthodes de recherche aléatoires de Monte-Carlo. La version déterministe de ces méthodes est la méthode de Quasi Monte-Carlo avec l'utilisation de suites à faible discrépance. Niederreiter (1988) montre qu'en utilisant ces suites, il est possible de borner l'erreur d'estimation commise. Cependant cette borne n'est plus dépendante de la discrépance de la suite, mais est dépendante d'un autre critère : la dispersion. Dans ce chapitre, nous présentons ce nouveau critère d'« uniformité ».

Dans la suite de cette thèse, nous notons $d$ la distance euclidienne.
La dispersion, ou *cover radius* en anglais, de la séquence $x = \{x_1, \ldots, x_n\}$ dans l'hypercube unité $I^s$ est définie par :

$$\delta(x) = \sup_{y \in I^s} \min_{i=1,\ldots,n} d(y, x_i) \tag{4.1}$$

La dispersion correspond au rayon de la plus grande boule vide de l'espace. Une illustration est présentée à la figure 4.1.

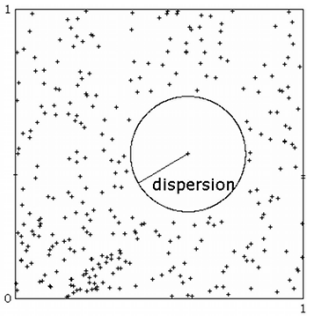



**FIGURE 4.1:** *Représentation de la dispersion d'une suite (en dimension 2) : la dispersion est le rayon de la plus grande boule de vide de l'espace.*

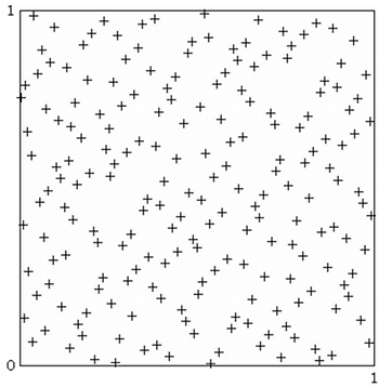
**Borne de la dispersion d'une suite :** Niederreiter (1988) montre que, pour toute suite à $n$ termes, la dispersion est bornée inférieurement par :

$$\delta(x) \geq \frac{1}{2\lfloor \sqrt[s]{n} \rfloor}. \tag{4.2}$$

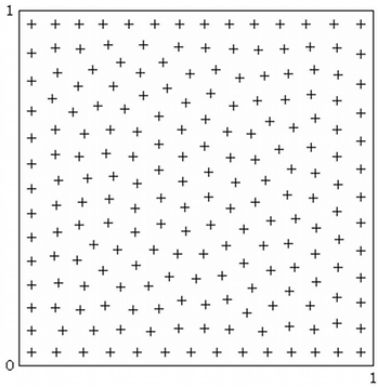
De plus, il montre que pour chaque dimension $s$ finie, il existe une suite $x$ de $I^s$ telle que $\lim_{n \to +\infty} \sqrt[s]{n}\delta(x) = \frac{1}{log(4)}$.
Par conséquent, pour chaque dimension $s$, il existe au moins une séquence $x$ dans $I^s$ telle que $\delta(x) = \mathcal{O}\left(\frac{1}{\sqrt[s]{n}}\right)$. En considérant l'inégalité 4.2 celle-ci est la plus petite possible.

**Discrépance et dispersion ne sont pas des critères équivalents :** à première vue, discrépance et dispersion peuvent être vue comme des critères équivalents. Ces critères sont équivalents lorsque la taille de la suite considérée est infinie. Cependant, lorsque la taille de celle-ci est fixe, ils ne sont absolument pas équivalents. Pour s'en convaincre, il suffit de regarder comment ils évoluent lorsque l'on rajoute un point à une suite : la dispersion de la suite ne peut que diminuer, tandis que la discrépance peut diminuer ou augmenter.
Un autre moyen de s'en convaincre consiste à regarder les configurations qui minimisent la dispersion et la discrépance. Pour une taille de suite adéquate, la configuration des points qui minimise la dispersion est la grille régulière. Or, nous avons vu précédemment à la section 2.1.3 que les grilles ne sont pas à discrépance faible. Par conséquent, discrépance et dispersion n'admettent pas les mêmes configurations optimales et sont donc différentes.

Afin d'illustrer graphiquement la différence entre faible discrépance et faible dispersion, nous avons représentés à la figure 4.2(a) une suite de Halton à faible discrépance à 200 points en dimension 2 (dont la dispersion estimée est égale à 0,101). Avec un algorithme de réduction de dispersion décrit par Gandar *et al.* (2011) et dans le paragraphe 5.4 de cette thèse, nous avons bougé les points de cette suite de manière à réduire sa dispersion. La figure 4.2(b) représente le résultat final : sa dispersion est égale à 0,054. Notons que les points ont tendance à former une grille régulière, qui, rappelons-le, n'est pas à faible discrépance.




(a) Suite à discrépance faible de Halton à 200 points en dimension 2 - dispersion = 0,101.




(b) Suite de Halton modifiée par l'algorithme de réduction de la dispersion de Gandar *et al.* (2011) - dispersion = 0,054. On peut remarquer une tendance des points à être disposés selon une grille régulière qui n'est pas à faible discrépance.

**FIGURE 4.2:** *Discrépance et dispersion ne sont pas deux critères équivalents.*

**La grille de Shukarev : une configuration qui, pour une taille adéquate, minimise la dispersion :** lorsque le nombre de points est égal à un entier à la puissance de la dimension, la configuration qui minimise la dispersion est la grille de Shukarev.
Les grilles classiques sont construites sur $I^s$ en plaçant l'« origine » de la grille au point $(0,0,\ldots,0)$. Pour une grille à $k$ points par dimension, chaque coordonnée des points est de la forme $\mu * \frac{1}{k-1}$ où $\mu \in \{0,\ldots,k-1\}$, *i.e.* les coordonnées sont des multiples $\frac{1}{k-1}$ d'ordre compris entre 0 et $k-1$.
La grille de Sukharev est une grille qui ne comporte pas de points sur les arêtes de l'hyper-cube. Pour une grille à $k$ points par dimension, chaque coordonnée des points est de la forme $\mu * \frac{1}{k-1} + \frac{1}{2k}$.

Les dispersions de ces deux grilles ne sont pas identiques, *e.g.*, pour une grille de 9 points en dimension 2, la dispersion d'une grille régulière est de $0,3536$ alors que celle de la grille de Shukarev est de $0,2357$. Cette différence peut paraître petite, cependant, en grande dimension, celle-ci prend une grande importance. Par exemple, en dimension $s = 10$, une grille de Shukarev avec $2^{10} = 1\,024$ points fournit la même qualité de dispersion qu'une grille classique de $3^{10} = 59\,049$ points.

Une représentation de la grille de Shukarev en dimension 2 avec 36 points est réalisée à la figure 4.3. Dans la suite de ce manuscrit, sauf mention contraire, lorsque nous utiliserons le terme grille ou grille régulière, nous ferons allusion aux grilles de Shukarev.

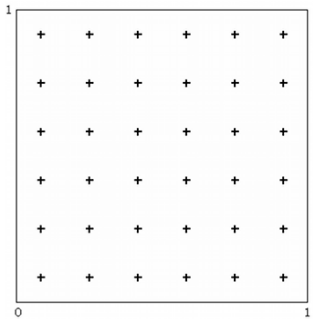



**FIGURE 4.3:** *Grille de Sukharev à 36 points dans $I^2$ : cette grille est la configuration qui minimise la dispersion lorsque la taille de la suite est appropriée.*

**Comment générer une suite à faible dispersion ?** Lorsque la taille de la suite est adéquate, nous avons vu au paragraphe précédent que la configuration qui minimise la dispersion est la grille de Shukarev. Cependant cette configuration nécessite un nombre de points qui croît exponentiellement avec la dimension. Disposer d'un tel nombre de points est en général difficile en pratique : il est donc nécessaire de pouvoir générer des suites de n'importe quelle taille. La génération de telles suites n'est en général pas aisée. Afin de résoudre ce problème, nous présentons dans le chapitre les différents algorithmes utilisés dans la littérature pour générer une suite à faible dispersion et nous proposons un nouvel algorithme.

**Comment estimer la dispersion d'une suite ?** L'estimation de la dispersion d'une suite n'est pas aisée. D'une manière générale, il n'existe pas de résultats théoriques sur la dispersion d'une suite à l'exception des suites de Hammersley, des grilles régulières ou des suites que l'on peut construire nous-mêmes. Dans cette partie, nous présentons à la fois des résultats théoriques et des méthodes numériques.

Les dispersions théoriques des grilles régulières sont connues : pour des suites à $k$ points par dimension en dimension $s$, la dispersion d'une grille régulière classique est de $\dfrac{\sqrt{s}}{2(k-1)}$ et celle d'une grille de Shukarev est de $\dfrac{\sqrt{s}}{2k}$. Peart et Mitchell (1992) proposent une méthode pour calculer théoriquement la dispersion d'une suite. Celle-ci se base sur la décomposition en cellules de Voronoï de l'espace $I^s$ avec, aux centres des cellules, les points de la suite. La construction des cellules de Voronoï par l'algorithme de Edelsbrunner (1987) se réalise avec une vitesse de l'ordre de $\mathcal{O}(n\log(n))$ lorsque $s \leq 2$, et en $\mathcal{O}(n^p)$ où $p = \left\lceil \dfrac{1}{2}(s+1) \right\rceil$, le plus grand entier inférieur ou égal à $\frac{1}{2}(s+1)$. Cette méthodologie est cependant difficile à appliquer dans le cas général, et, à notre connaissance, seules les dispersions des suites de Hammersley (voir Peart, 1982) ont été calculées.

A défaut de disposer d'outils théoriques puissants pour calculer la dispersion, il est possible de l'estimer de manière numérique. Une manière simple consiste à générer sur $I^s$ une grille régulière (classique) et de calculer pour chaque point de la grille la distance minimale avec les points de la suite. La dispersion est la maximale de ces distances sur tous les points de la grille. La qualité de cette estimation est fortement dépendante de la résolution de la grille utilisée : plus la résolution est importante, meilleure sera l'estimation.
Cependant, pour des estimations de bonne qualité, ou lorsque la dimension est élevée, cela requiert un nombre de calculs très important, voire souvent inaccessible sur un ordinateur classique. Nous pouvons limiter le nombre de calculs en sous-divisant le problème d'estimation en plusieurs sous-problèmes. Pour cela, nous allons découper uniformément chaque dimension de l'hypercube unité en $k$ segments, obtenant ainsi $k^s$ sous-cubes. Nous devons ensuite identifier à quel sous-cube appartiennent chaque point de la suite ainsi que l'ensemble des $3^s - 1$ cubes voisins de chaque sous-cube. Ensuite, nous munissons chaque sous-cube d'une grille régulière (classique) et pouvons calculer la dispersion sur cette grille avec les points de la suite du sous-cube et des sous-cubes voisins. La dispersion finale sera la dispersion maximale évaluée sur l'ensemble des sous-cubes.
Nous utilisons cette méthode pour estimer les dispersions des suites utilisées dans les expériences numériques.

## 4.2 Retour d'expérience sur la dispersion et l'erreur de généralisation en classification

Dans le chapitre précédent, à la section 3.3, nous avons montré que la discrépance n'est pas un critère adéquat pour générer les premiers points d'apprentissage en classification : nous obtenions de meilleurs résultats expérimentaux sur des grilles régulières que sur des suites à discrépance faible. Nous nous interrogions alors sur une caractérisation, autre que la discrépance, qui permettrait d'expliquer le résultat obtenu sur les grilles et sur les suites à faible discrépance. Dans ce chapitre, nous avons présenté une nouvelle définition de l'« uniformité » d'une suite : la dispersion. Dans cette section, nous revenons sur les expériences numériques précédentes en prenant le point de vue de la dispersion.

La table 4.1 fournit les dispersions des suites utilisées dans les expériences précédentes. Nous pouvons ainsi remarquer que les erreurs en apprentissage élevées sont corrélées avec les dispersions élevées des échantillons d'apprentissage.

En comparant les tables 3.1 et 3.2 à la table 4.1, nous pouvons remarquer que les grilles régulières, qui obtiennent les meilleurs résultats en erreur en généralisation, possèdent également la dispersion la plus faible. De plus, pour une dimension fixée, cette différence est plus significative pour les échantillons de petite taille. Enfin, nous pouvons remarquer également que la suite de Halton se caractérise par une dispersion plus faible que celle de la suite de Sobol pour toutes nos expériences [1] et possède souvent un meilleur rang que la suite de Sobol.

La dispersion des points apparaît donc comme une caractérisation de l'« uniformité » des points d'apprentissage plus adaptée à la classification et permet d'expliquer les résultats obtenus en comparant les apprentissages sur grilles et sur suites à faible discrépance. A partir de ces résultats, nous pouvons donc émettre l'hypothèse selon laquelle minimiser la dispersion est une stratégie adaptée pour réduire l'erreur en généralisation en classification.

1. Exception faite des échantillons de taille 2 500 en dimension 2, de taille 3 375 en dimension 3 et de taille 7 776 en dimension 5.

| Dimension | Taille de l'échantillon | Grille | Suite de Halton | Suite de Sobol |
|---|---|---|---|---|
| 2 | 100 | **7,9** | 12,7 | 13,8 |
| | 225 | **5,0** | 8,4 | 8,9 |
| | 400 | **3,7** | 7,5 | 8,9 |
| | 676 | **2,8** | 4,9 | 5,2 |
| | 2 500 | **1,4** | 2,9 | 2,4 |
| 3 | 64 | **28,9** | 32,4 | 36,3 |
| | 512 | **12,4** | 15,5 | 18,3 |
| | 1 000 | **9,6** | 14,0 | 14,9 |
| | 3 375 | **6,2** | 10,7 | 8,9 |
| 4 | 625 | **22,2** | 30,2 | 30,4 |
| | 1 296 | **17,8** | 22,2 | 25,3 |
| | 2 401 | **16,7** | 21,0 | 25,2 |
| | 4 096 | **14,3** | 16,4 | 19,2 |
| | 6 561 | **12,5** | 16,4 | 17,8 |
| | 10 000 | **11,1** | 14,6 | 15,3 |
| 5 | 243 | **37,3** | 44,8 | 56,4 |
| | 1 024 | **28,0** | 34,6 | 43,3 |
| | 3 125 | **22,4** | 28,1 | 31,4 |
| | 7 776 | **18,6** | 27,1 | 25,1 |

**TABLE 4.1:** *Dispersion des échantillons d'apprentissage utilisés dans les expériences numériques de la section 3.3 ($*10^{-2}$). Les dispersions les plus faibles sont écrites en caractère gras et correspondent aux grilles régulières.*

## 4.3 Un lien particulier entre dispersion et erreur de généralisation en classification

Le but de cette section est d'établir un lien entre erreur en généralisation et dispersion des points d'apprentissage. Dans un premier temps, nous établissons un lien théorique en utilisant un algorithme d'apprentissage similaire à celui des $k$-PPV ($k$-plus proches voisins). Dans une seconde partie, nous illustrons ce résultat sur trois problèmes jouets d'apprentissage de variétés.

### 4.3.1 Lien théorique entre dispersion et erreur d'apprentissage pour une procédure simple de classification

Soit $f$ une fonction définie sur $I^s$ à valeurs dans $\{-1,+1\}$. Nous cherchons à approximer cette fonction à partir d'un ensemble d'apprentissage $E$ de dispersion $\delta$. Notons également $B(x,R)$ la boule de centre $x$ et de rayon $R$. De plus, nous supposons que $f$ possède la propriété de régularité suivante :
pour tout $x \in I^s$, il existe une boule $B(x_0,R)$ centrée en un point $x_0$ de rayon $R$ telle que :
- $x \in B(x_0,R)$ ;
- $f$ est constante sur $B(x_0,R)$.

Mathématiquement, cela se traduit par : $\exists R > 0$ tel que :
- $\forall x \in I^s, \exists x_0 \in I^s | x \in B(x_0,R)$ ;
- $\forall y \in B(x_0,R)\, f(y) = f(x)$.

A première vue, cette propriété de régularité peut paraître assez restrictive. Cependant, en pratique, la majorité des classes de fonctions étudiées respectent cette propriété de régularité. La constante $R$ peut être estimée à partir du rayon de courbure de la fonction cible. Les figures 4.4(a), 4.5(a) 4.6(a) illustrent cette classe de fonctions en dimension 2.

**Théorème.** *Lien entre l'erreur en généralisation et la dispersion*
*Soit $E^+$ (resp. $E^-$) l'ensemble des points d'apprentissage $x_i$ de $E$ tels que $f(x_i) = +1$ (resp. $f(x_i) = -1$), soit $\mathscr{A}$ l'algorithme d'apprentissage approximant la fonction $f$ par $\mathscr{A}(E) = \hat{f}$ tel que :*

$$\hat{f}(x) = \begin{cases} +1 & si\ \forall x_i^- \in E^-, d(x_i^-, x) > 2\delta. \\ -1 & si\ \forall x_i^+ \in E^+, d(x_i^+, x) > 2\delta. \\ aléatoire & sinon \end{cases}$$

*Alors, il existe une constante $\lambda > 0$ telle que, pour tout ensemble d'apprentissage $E$ de dispersion $\delta < R$, l'algorithme $\mathscr{A}$ délivre une approximation de $f$ possédant une erreur en généralisation $\mathscr{R}[\mathscr{A}(E)]$ satisfaisant :*

$$\mathscr{R}[\mathscr{A}(E)] < \lambda\delta.$$

*Démonstration.* Notons $\chi_{f^+} = \{x \in I^s | f(x) = +1\}$ l'ensemble des points de l'espace étiquetés $+1$, et $\chi_{f^-} = \{x \in I^s | f(x) = -1\}$ l'ensemble des points de l'espace étiquetés $-1$. Notons également les ensembles suivants : $F^+ = \{x \in I^s | \forall x_i^- \in E^-, d(x_i^-, x) > 2\delta\}$ et $F^- = \{x \in I^s | \forall x_i^+ \in E^+, d(x_i^+, x) > 2\delta\}$. Enfin, définissons la distance d'un point $x \in I^s$ à un ensemble fini $S \subset I^s$, comme le minimum des distance $d(x,y)$, où $y$ prend ses valeurs dans $S$.

1. Les ensembles $F^+$ et $F^-$ sont disjoints.
   En effet, $x \in F^+ \Rightarrow \forall x^- \in E, d(x, x^-) > 2\delta$. Or $\exists x' \in E | d(x,x') \leq \delta$, par définition de $\delta$. Par conséquent, nécessairement $x' \in \chi_{f^+}$, car le point de label négatif le plus proche est à une distance supérieure que $\chi_{f^+}$. A fortiori, $\exists x_0 \in I^s$ tel que $x' \in B(x_0, \delta)$ et $B(x_0, \delta) \subset \chi_{f^+}$ car $R > \delta$. Ainsi, $x \notin F^-$. Avec un raisonnement analogue, nous pouvons démontrer également que $x \in F^- \Rightarrow x \notin F^+$.

2. Prouvons maintenant que $F^+ \subset \chi_{f^+}$. Soit $x \in F^+$. Supposons que $x \in \chi_{f^-}$. La condition de régularité sur $f$ implique que : $\exists x_0 \in \chi_{f^-} | x \in B(x_0, R)$ et $B(x_0, R) \subset \chi_{f^-}$. A *fortiori* $\exists x' \in \chi_{f^-} | x \in B(x', \delta)$ et $B(x', \delta) \subset \chi_{f^-}$, car $R > \delta$. Par définition de la dispersion, $\exists x_i \in E$, tel que $x_i \in B(x', \delta) \subset \chi_{f^-}$. Par conséquent, $d(x, x_i) < 2\delta$ : ceci est en contradiction avec l'hypothèse ($x \in F^+$). Par conséquent, $x \in \chi_{f^+}$, car $F^+ \subset \chi_{f^+}$. De façon similaire, nous pouvons montrer que $F^- \subset \chi_{f^-}$. En d'autres termes, l'algorithme d'apprentissage ne fait pas d'erreurs sur $F^+$, ni sur $F^-$.

3. Utilisons maintenant le premier résultat pour borner l'erreur en généralisation définie par $\mathscr{R}[A(E)] = \int_{I^s} |f - \hat{f}|(x)dx$. Sur $F^+$ et $F^-$, les fonctions $f$ et $\hat{f}$ sont égales, par conséquent les erreurs sont localisées dans l'ensemble $I^s - F^+ - F^-$. Ainsi, $\mathscr{R}[A(E)] = \int_{I^s - F^+ - F^-} |f - \hat{f}|(x)dx$, impliquant alors $\mathscr{R}[A(E)] < 2\mathbf{V}(I^s - F^+ - F^-)$, où $\mathbf{V}$ est le volume de cet ensemble. Soient $\partial f$ la frontière entre $\chi_{f^-}$ et $\chi_{f^+}$, et $M = \{x \in I^s | d(x, \partial f) \leq 2\delta\}$ .
   On a clairement $I^s - F^+ - F^- \subset M$. En effet, $x \notin F^+$ implique $d(x, E^-) \leq 2\delta$, ce qui implique $d(x, \chi_{f^-}) \leq 2\delta$, car $E^- \subset \chi_{f^-}$. De façon similaire, nous avons $d(x, \chi_{f^+}) \leq 2\delta$.
   Par conséquent, $\mathscr{R}[A(E)] \leq 2\mathbf{V}(M)$. Cependant, $\mathbf{V}(M) \leq 4\delta\mathbf{S}(\partial f)$, où $\mathbf{S}(\partial f)$ est l'intégrale de la surface [2] $\partial f$. La condition de régularité de la fonction $f$ implique que cette intégrale est finie.

□

2. Ou sous-variété.

Pour conclure, avec cet algorithme particulier d'apprentissage, et sous les conditions de régularité de la frontière de classification, l'erreur en généralisation décroît linéairement avec la dispersion de l'ensemble d'apprentissage. Ceci suggère que la dispersion est un indicateur pertinent de mesure de la qualité d'un ensemble d'apprentissage en apprentissage de variétés, ou classification.

### 4.3.2 Illustrations expérimentales du théorème

Le but de cette section est d'illustrer expérimentalement le théorème précédent. Dans cette optique, nous étudions l'effet de la dispersion sur l'erreur en généralisation sur trois variétés jouets en dimension 2. Nous commençons par décrire le protocole d'apprentissage utilisé dans cette illustration expérimentale, puis nous présentons trois variétés jouets, à savoir la variété des deux cercles, la variété du sinus et une variété mixte des deux dernières.

#### 4.3.2.1 Protocole d'apprentissage utilisé

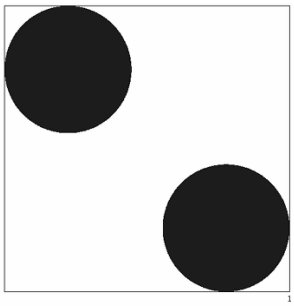
Pour chaque variété, nous avons calculé la constante de régularité $R$. Utilisant l'hypothèse du théorème selon laquelle $\delta \geq R$ et l'équation 4.2, nous pouvons estimer le nombre minimal de points nécessaires pour satisfaire l'hypothèse du théorème.
Nous tirons une suite aléatoire uniforme de taille adéquate et nous présentons ces points à l'oracle expert. Nous estimons alors la fonction cible avec l'algorithme d'apprentissage du théorème et nous estimons l'erreur en généralisation avec une grille régulière classique de 38 809 points et en utilisant la fonction de coût classique. Puis nous réduisons la dispersion de la suite en appliquant plusieurs itérations de l'algorithme de réduction de la dispersion présenté à la section 5.4. A chaque itération, nous présentons la nouvelle suite à l'oracle expert et apprenons la variété avec l'algorithme d'apprentissage précédent. Ce processus est répété jusqu'à l'obtention d'une séquence ayant la plus petite dispersion. Finalement, nous représentons l'évolution de l'erreur en généralisation en fonction de la dispersion.

#### 4.3.2.2 Variété jouet des deux cercles

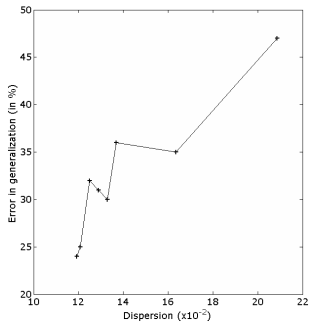
La première variété étudiée est la variété des deux cercles, dont la fonction cible est alors définie par :

$$\begin{array}{rcl} f : I^2 & \rightarrow & \{-1,1\} \\ & & \left\{ \begin{array}{ll} 1 & \text{si } (x_1,x_2) \in B((0.3,0.7),0.2) \\ & \text{ou } (x_1,x_2) \in B((0.8,0.2),0.2) \\ -1 & \text{sinon} \end{array} \right. \\ (x_1,x_2) & \mapsto & \end{array}$$

Une représentation graphique de la fonction cible est réalisée à la figure 4.4(a). La constante $R$ est bornée par 0.15 qui est la distance minimale entre les deux cercles. Par conséquent, le nombre minimal de points requis est de 12. Les résultats expérimentaux des apprentissages réalisés sont présentés à la figure 4.4(b). Nous pouvons y voir clairement que l'erreur en généralisation est bornée supérieurement par une fonction linéaire de la dispersion ; et plus la dispersion est élevée, plus l'erreur en généralisation est élevée également.



(a) Variété des deux cercles.




(b) Effet de la dispersion de l'ensemble d'apprentissage sur l'erreur en généralisation.

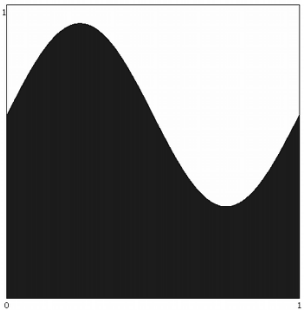
**FIGURE 4.4:** *Effet de la dispersion sur l'apprentissage de la variété des deux cercles en dimension 2. Apprentissage réalisé avec 12 points.*

#### 4.3.2.3 Variété du sinus

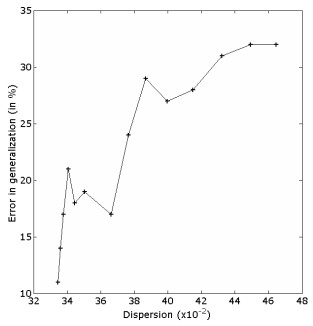
La deuxième variété étudiée est la variété du sinus, dont la fonction cible est alors définie par :

$$\begin{array}{rcl} f : I^2 & \rightarrow & \{-1,1\} \\ (x_1,x_2) & \mapsto & \left\{ \begin{array}{ll} 1 & \text{si } x_2 > \frac{1}{2}\sin(2\pi x_1) + \frac{1}{2} \\ -1 & \text{sinon} \end{array} \right. \end{array}$$

Une représentation graphique de la fonction cible est réalisée à la figure 4.5(a). La constante $R$ peut être calculée par le rayon de courbure de la fonction analytique. Le rayon minimal du disque maximal de même couleur est égal à 0.07 et est positionné dans les coudes de la courbe frontière. Par conséquent, le nombre minimal de points requis est de 50. Les résultats expérimentaux des apprentissages réalisés sont présentés à la figure 4.5(b). Nous pouvons y voir clairement que l'erreur en généralisation est bornée supérieurement par une fonction linéaire de la dispersion ; et plus la dispersion est élevée, plus l'erreur en généralisation est élevée également.



(a) Variété du sinus.




(b) Effet de la dispersion de l'ensemble d'apprentissage sur l'erreur en généralisation.

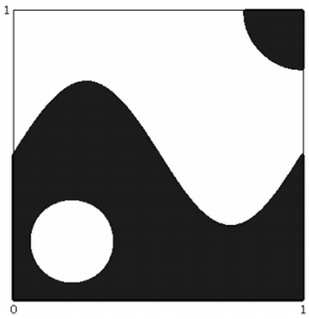
**FIGURE 4.5:** *Effet de la dispersion sur l'apprentissage de la variété du sinus en dimension 2. Apprentissage réalisé avec 50 points.*

#### 4.3.2.4 Variété du sinus et des deux cercles

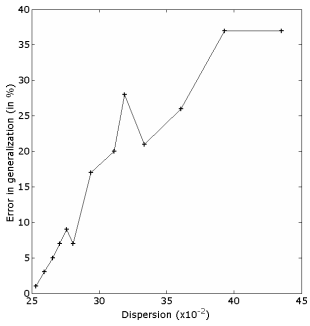
La troisième et dernière variété étudiée est une combinaison des deux précédentes. Cette variété du sinus et des cercles possède alors la fonction suivante pour la fonction cible :

$$
\begin{array}{rcl}
f : I^2 & \to & \{-1,1\} \\
(x_1,x_2) & \mapsto & \left\{ \begin{array}{ll}
1 & \text{si } (x_1,x_2) \in B((0.2,0.2),0.15) \\
-1 & \text{si } (x_1,x_2) \in B((1,1),0.2) \\
1 & \text{si } x_2 > \frac{1}{2}\sin(2\pi x_1) + \frac{1}{2} \\
 & \text{et } (x_1,x_2) \notin B((0.2,0.2),0.15) \\
-1 & \text{si } x_2 < \frac{1}{2}\sin(2\pi x_1) + \frac{1}{2} \\
 & \text{et } (x_1,x_2) \notin B((1,1),0.2)
\end{array} \right.
\end{array}
$$

Une représentation graphique de la fonction cible est réalisée à la figure 4.6(a). La constante $R$ peut être calculée par le rayon de courbure de la fonction analytique. Le rayon minimal du disque maximal de même couleur est égal à 0.07 et est positionné dans les coudes de la courbe frontière. Par conséquent, le nombre minimal de points requis est de 50. Les résultats expérimentaux des apprentissages réalisés sont présentés à la figure 4.6(b). Nous pouvons y voir clairement que l'erreur en généralisation est bornée supérieurement par une fonction linéaire de la dispersion ; et plus la dispersion est élevée, plus l'erreur en généralisation est élevée également.




(a) Variété du sinus et des cercles.




(b) Effet de la dispersion de l'ensemble d'apprentissage sur l'erreur en généralisation.

**FIGURE 4.6:** *Effet de la dispersion sur l'apprentissage de la variété du sinus et des cercles en dimension 2. Apprentissage réalisé avec 50 points.*

**Conclusion :** nous avons illustré sur ces trois exemples jouets d'apprentissage de variétés satisfaisant l'hypothèse de régularité que l'erreur en généralisation est bien fonction de la dispersion des points d'apprentissage. De plus, dans ce cas particulier, cette dépendance est linéaire.
La question naturelle qui en découle maintenant est de savoir si, sans cette hypothèse forte de régularité, l'erreur en généralisation est aussi dépendante de la dispersion.

## 4.4 La dispersion : un critère adéquat pour générer des points d'apprentissage d'études de variétés ?

Dans les sections précédentes, nous avons mis en évidence que la discrépance n'est pas adaptée à l'apprentissage de variétés et nous avons conjoncturé sur le fait que la minimisation de la

dispersion est un critère adapté. Nous l'avons démontré théoriquement dans un cas particulier d'apprentissage. Nous avons illustré ce point en respectant des conditions de régularité de la variété. Dans cette section, nous essayons de nous libérer de cette condition.

De façon similaire à la section 3.3, nous avons généré un ensemble artificiel de règles de classification comme présenté à l'annexe B, page 129. Pour chaque règle, nous apprenons avec deux échantillons différents de même taille et trois méthodes d'apprentissage. Les premiers échantillons sont des suites aléatoires uniformes. Aucune de ces séquences ne possède un nombre de points tel que l'on puisse utiliser une grille de Shukarev, qui minimise la dispersion. Ainsi, à partir de ces suites, nous avons réduit la dispersion de la séquence en utilisant l'algorithme décrit par Gandar *et al.* (2011). Nous avons présenté chaque échantillon à l'oracle expert pour étiquetage des points, puis nous avons réalisé l'apprentissage, et enfin, nous avons estimé la qualité de celui-ci.

Les algorithmes d'apprentissage que nous avons utilisés sont les $k$-NN, l'algorithme des SVMs et les arbres de classification avec les configurations suivantes :

- $k$-NN : validation croisée sur 5 échantillons basée sur l'erreur d'apprentissage (pour chaque règle et chaque échantillonnage) pour choisir la taille de voisinage dans l'ensemble $\{3,5,7\}$.
- SVM : validation croisée sur 5 échantillons basée sur l'erreur d'apprentissage (pour chaque règle et chaque échantillonnage) pour choisir la largeur de bande $\sigma$ du noyau gaussien entre les valeurs 0.2 et 2. La paramètre $C$ est fixé [3] à 10 000.
- arbre : nous divisons les nœuds de l'arbre qui possèdent 3 individus ou plus.

La table 4.2 présente les résultats obtenus sur l'erreur en généralisation sue 1 000 problèmes d'apprentissage. L'erreur est estimée sur une séquence régulière de 6 000 points en utilisant la fonction de coût classique. Pour chaque ensemble de problèmes, et pour chaque méthode d'apprentissage, nous avons également estimé la différence moyenne entre l'erreur en généralisation obtenue sur des échantillons aléatoires et sur des échantillons à dispersion minimale. Nous présentons également un intervalle de confiance de cette différence moyenne avec un degré de confiance de 99%.

3. Nos problèmes sont « hard-margin ».

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Suite aléatoire | Suite de dispersion minimale | Différence moyenne | Intervalle de confiance à 99 % de la différence moyenne | |
|---|---|---|---|---|---|---|---|
| | | | | | | Borne inférieure | Borne supérieure |
| 2 | 200 | SVM | 6,46 | **2,89** | -3,57 | -4,06 | -3,08 |
| | | kNN | 3,94 | **2,32** | -1,61 | -1,69 | -1,54 |
| | | Arbre | 5,21 | **4,13** | -1,08 | -1,21 | -0,96 |
| | 450 | SV M | 4,30 | **1,74** | -2,56 | -2,84 | -2,28 |
| | | kNN | 2,59 | **1,45** | -1,15 | -1,20 | -1,10 |
| | | Arbre | 3,51 | **2,84** | -0,67 | -0,74 | -0,60 |
| | 800 | SVM | 3,10 | **1,25** | -1,85 | -2,04 | -1,66 |
| | | kNN | 1,95 | **1,08** | -0,87 | -0,91 | -0,84 |
| | | Arbre | 2,69 | **2,19** | -0,49 | -0,54 | -0,45 |
| | 1 000 | SVM | 2,70 | **1,07** | -1,63 | -1,79 | -1,48 |
| | | kNN | 1,77 | **0,95** | -0,82 | -0,85 | -0,79 |
| | | Arbre | 2,41 | **1,94** | -0,47 | -0,51 | -0,43 |
| | 1 500 | SVM | 2,09 | **0,79** | -1,25 | -1,41 | -1,18 |
| | | kNN | 1,41 | **0,74** | -0,68 | -0,70 | -0,65 |
| | | Arbre | 1,94 | **1,58** | -0,36 | -0,39 | -0,33 |
| | 2 000 | SVM | 1,67 | **1,74** | -1,06 | -1,15 | -0,98 |
| | | kNN | 1,23 | **0,64** | -0,58 | -0,61 | -0,56 |
| | | Arbre | 1,72 | **1,38** | -0,34 | -0,36 | -0,31 |

**TABLE 4.2:** *Moyenne de l'erreur en généralisation estimée sur 1 000 problèmes de classification (en %). Pour chaque problème et chaque méthode d'apprentissage, les meilleurs résultats sont écrits en gras.*

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Suite aléatoire | Suite de dispersion minimale | Différence moyenne | Intervalle de confiance à 99 % de la différence moyenne | |
|---|---|---|---|---|---|---|---|
| | | | | | | Borne inférieure | Borne supérieure |
| 3 | 50 | SVM | 12,13 | **9,82** | -2,31 | -2,97 | -1,65 |
| | | kNN | 12,48 | **8,88** | -3,59 | -3,82 | -3,37 |
| | | Arbre | 16,41 | **12,45** | -3,97 | -4,35 | -3,58 |
| | 200 | SVM | 9,85 | **5,20** | -4,64 | -5,13 | -4,16 |
| | | kNN | 7,30 | **5,02** | -2,28 | -2,37 | -2,19 |
| | | Arbre | 10,44 | **8,94** | -1,49 | -1,67 | -1,33 |
| | 450 | SVM | 7,33 | **3,54** | -4,64 | -5,13 | -4,16 |
| | | kNN | 5,32 | **3,57** | -1,74 | -1,79 | -1,68 |
| | | Arbre | 7,78 | **6,69** | -1,09 | -1,20 | -0,99 |
| | 800 | SVM | 5,69 | **2,71** | -2,98 | -3,21 | -2,73 |
| | | kNN | 4,31 | **2,85** | -1,45 | -1,49 | -1,41 |
| | | Arbre | 6,49 | **5,58** | -0,90 | -0,98 | -0,83 |
| | 1 500 | SVM | 4,09 | **2,00** | -2,08 | -2,24 | -1,93 |
| | | kNN | 3,43 | **2,22** | -1,21 | -1,24 | -1,17 |
| | | Arbre | 5,27 | **4,63** | -0,63 | -0,68 | -0,58 |
| | 2 000 | SVM | 3,56 | **1,76** | -1,80 | -1,92 | -1,67 |
| | | kNN | 3,10 | **1,99** | -1,10 | -1,13 | -1,07 |
| | | Arbre | 4,81 | **4,21** | -0,61 | -0,64 | -0,56 |
| | 2 500 | SVM | 3,09 | **1,54** | -1,53 | -1,65 | -1,46 |
| | | kNN | 2,83 | **1,79** | -1,04 | -1,06 | -1,00 |
| | | Arbre | 4,37 | **3,83** | -0,53 | -0,57 | -0,49 |

**TABLE 4.2:** *Moyenne de l'erreur en généralisation estimée sur 1 000 problèmes de classification (en %). Pour chaque problème et chaque méthode d'apprentissage, les meilleurs résultats sont écrits en gras.*

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Suite aléatoire | Suite de dispersion minimale | Différence moyenne | Intervalle de confiance à 99 % de la différence moyenne | |
|---|---|---|---|---|---|---|---|
| | | | | | | Borne inférieure | Borne supérieure |
| 4 | 50 | SVM | 13,70 | **12,98** | -0,71 | -1,25 | -0,17 |
| | | kNN | 15,95 | **12,08** | -3,85 | -4,09 | -3,62 |
| | | Arbre | 20,70 | **16,39** | -4,32 | -4,71 | -3,91 |
| | 200 | SVM | 10,91 | **6,88** | -4,03 | -4,47 | -3,59 |
| | | kNN | 9,85 | **7,01** | -2,76 | -2,85 | -2,66 |
| | | Arbre | 14,30 | **12,36** | -1,94 | -2,15 | -1,73 |
| | 450 | SVM | 9,92 | **4,99** | -4,93 | -5,30 | -4,56 |
| | | kNN | 7,73 | **5,50** | -2,23 | -2,19 | -2,16 |
| | | Arbre | 11,73 | **10,49** | -1,23 | -1,36 | -1,10 |
| | 800 | SVM | 8,21 | **3,92** | -4,29 | -4,59 | -4,00 |
| | | kNN | 6,44 | **4,53** | -1,90 | -1,95 | -1,85 |
| | | Arbre | 10,03 | **9,07** | -0,96 | -1,06 | -0,86 |
| | 1 000 | SVM | 7,61 | **3,66** | -3,95 | -4,21 | -3,72 |
| | | kNN | 7,69 | **4,27** | -1,80 | -1,85 | -1,76 |
| | | Arbre | 9,57 | **8,67** | -0,81 | -0,90 | -0,72 |
| | 1 500 | SVM | 6,32 | **3,12** | -3,21 | -3,39 | -3,01 |
| | | kNN | 5,32 | **3,75** | -1,58 | -1,63 | -1,55 |
| | | Arbre | 8,61 | **7,84** | -0,77 | -0,84 | -0,70 |
| | 2 000 | SVM | 5,57 | **2,79** | -2,78 | -2,93 | -2,64 |
| | | kNN | 4,94 | **3,42** | -1,52 | -1,56 | -1,49 |
| | | Arbre | 8,03 | **7,37** | -0,77 | -0,71 | -0,59 |

**TABLE 4.2:** *Moyenne de l'erreur en généralisation estimée sur 1 000 problèmes de classification (en %). Pour chaque problème et chaque méthode d'apprentissage, les meilleurs résultats sont écrits en gras.*

| Dimension | Taille de l'échantillon | Méthode d'apprentissage | Suite aléatoire | Suite de dispersion minimale | Différence moyenne | Intervalle de confiance à 99 % de la différence moyenne | |
|---|---|---|---|---|---|---|---|
| | | | | | | Borne inférieure | Borne supérieure |
| 5 | 200 | SVM | 11,83 | **8,31** | -3,52 | -3,91 | -3,13 |
| | | kNN | 12,25 | **9,01** | -3,25 | -3,36 | -3,12 |
| | | Arbre | 17,68 | **15,36** | -2,32 | -2,57 | -2,08 |
| | 500 | SVM | 10,96 | **5,76** | -5,19 | -5,57 | -4,82 |
| | | kNN | 9,41 | **7,04** | -2,36 | -2,43 | -2,29 |
| | | Arbre | 14,69 | **12,93** | -1,76 | -1,92 | -1,59 |
| | 1 000 | SVM | 11,82 | **8,31** | -3,51 | -3,91 | -3,13 |
| | | kNN | 12,25 | **9,00** | -3,25 | -3,36 | -3,14 |
| | | Arbre | 17,68 | **15,36** | -2,32 | -2,57 | -2,08 |
| | 5 000 | SVM | 5,81 | **2,92** | -2,81 | -4,64 | -1,15 |
| | | kNN | 5,68 | **3,95** | -1,72 | -2,09 | -1,36 |
| | | Arbre | 10,07 | **9,30** | -0,77 | -1,36 | -0,18 |

**TABLE 4.2:** *Moyenne de l'erreur en généralisation estimée sur 1000 problèmes de classification (en %). Pour chaque problème et chaque méthode d'apprentissage, les meilleurs résultats sont écrits en gras.*

Nous pouvons remarquer que la moyenne de l'erreur en généralisation est supérieure pour les suites aléatoires à celle pour les suites à dispersion minimale. L'estimation de la différence moyenne par intervalles de confiance à degré de confiance à 99%, qui sont toutes négatives, renforce significativement ce résultat. Ce résultat est valable pour les trois types d'algorithmes d'apprentissage utilisés. Nous pouvons remarquer également que ce gain est supérieur lorsque les tailles d'échantillons sont petites.
Ces résultats ont été obtenus sur des fonctions cibles quelconques et avec des tailles d'échantillons variées, sans prendre en compte les hypothèses émises dans le théorème de la section précédente. Par conséquent, il apparaît que l'hypothèse de régularité de la fonction cible n'est pas nécessaire pour obtenir de bons résultats.

Pour conclure, ces expériences, couplées aux expériences sur la discrépance (voir les sections 3.3 et 4.2), montrent qu'il est clair que la faible dispersion fournit les meilleurs résultats dans l'apprentissage premier des problèmes de classification ou de variétés.

Enfin, Iwata et Ishii (2002) ont observé expérimentalement un gain en qualité de classification avec l'algorithme du réseau de neurones multi-couches en utilisant les suites à discrépance faible par rapport aux suites aléatoires. Ce résultat est en adéquation avec notre proposition. En effet, la dispersion d'une suite aléatoire uniforme est, en général, supérieure à la dispersion d'une suite à faible discrépance.

## 4.5 Conclusion

A première vue, la différence entre la classification et la régression peut apparaître comme surprenante. L'explication de cette différence réside dans la variation d'Hardy-Krause qui est utilisée pour borner la différence entre l'erreur en généralisation et l'erreur empirique en régression. Cette variation est la somme sur toutes les combinaisons possibles des directions de l'espace (voir équation A.2 à la page 126) de fonctions dérivées (voir équations A.3 & A.3 à la page 126). Par conséquent, de manière à capter correctement ces variations, il est essentiel de disposer d'un échantillon de points qui couvre l'ensemble de l'espace et dans toutes les directions possibles. Ainsi, vaut-il mieux éviter les structures régulières privilégiant les directions. Les suites à faible discrépance, qui ont de bonnes propriétés de projection sur chaque axe, représentent donc un bon compromis entre bonne distribution globale et absence de structures.

Dans le cas de la classification, la variation d'Hardy-Krause ne joue plus aucun rôle : elle est égale à l'infini dans la majorité des cas. Dans cette configuration d'apprentissage, le seul critère est de disposer partout d'une information proche. En effet, en classification, la fonction peut être vue comme constante localement. Par conséquent, un point est de même « label » qu'un point voisin avec une probabilité d'autant plus forte que ces points sont proches. Par conséquent, il est nécessaire de disposer en chaque endroit de l'espace d'une information ( *i.e.* point d'apprentissage) qui soit proche. Autrement dit, il est important de minimiser les ensembles convexes ne disposant pas d'information. Cela revient donc à minimiser la dispersion !

En apprentissage, à notre connaissance, l'identification et l'utilisation du critère de dispersion sont un acte nouveau : aucun travail ne propose explicitement ce critère dans la littérature. Néanmoins, cette notion commence petit à petit à se répandre au sein de la communauté d'apprentissage. Par exemple, dans un cadre applicatif en apprentissage par renforcement, Fonteneau *et al.* (2010a) et Fonteneau *et al.* (2010b) utilisent la notion de sparsité, pour étudier la convergence d'un algorithme. Ils définissent la sparsité (*sparsity* en anglais) comme étant : « *the sparsity can be seen as*

*the radius of the largest non-visited state space area* », et la traduisent mathématiquement par :

$$sparsity = \sup_{x \in \mathscr{X}} \min_{y \in \mathscr{Y}} \|x - y\| .$$

A y regarder de plus près, cela correspond exactement à la dispersion...
Enfin, dans un cadre plus théorique, Fonteneau *et al.* (2011) étudient les propriétés de généralisation dans le contexte *min max* d'apprentissage par renforcement sur des fonctions lipchitziennes à supports finis et à optimisation temporelle finie. Ils écrivent les équations de convergence en faisant également appel à la notion de sparsité. Sous l'hypothèse d'avoir un échantillonnage itératif qui diminue la sparsité, ils prouvent des convergences d'apprentissages.
Leurs travaux utilisent cette notion de dispersion, ou sparsité, pour caractériser des états d'un algorithme d'apprentissage par renforcement, mais ils ne présentent pas formellement de méthodes pour générer de tels plans. Cette question de génération de suite à faible dispersion est donc un verrou scientifique important dans différents domaines de recherche. Dans le chapitre suivant, nous présentons des éléments de réponse pour lever, au moins partiellement, ce verrou...

# - CHAPITRE 5 -

# GÉNÉRATION D'UNE SUITE À FAIBLE DISPERSION

## Sommaire



NOUS AVONS MIS EN ÉVIDENCE, dans le chapitre précédent, que la discrépance n'est pas le critère adéquat pour générer les premiers points d'apprentissage, et nous avons alors proposé le critère de la dispersion. Dans ce chapitre, nous faisons un état de l'art des différents algorithmes de génération de ces suites à faible dispersion de taille quelconque et présentons un nouvel algorithme.

Les travaux pionniers de Johnson *et al.* (1990) proposent deux critères pour générer une suite à faible dispersion. Le premier critère est celui dit du *minimax* qui n'est autre que le critère de dispersion. Celui-ci étant difficile à contrôler, un deuxième critère, basé sur la distance entre les points de la suite, est proposé : le critère du *maximin*.
Les algorithmes de génération de ces suites peuvent se baser sur l'un ou l'autre de ces critères. En plus de ceux-ci, les algorithmes peuvent être différenciés par la stratégie qu'ils utilisent : stratégie

d'ajout de points, stratégie de suppression de points ou stratégie d'échange de points. Enfin, certains algorithmes proposent une résolution déterministe du problème alors que d'autre délivrent une réponse stochastique obtenue en utilisant des processus de génération aléatoire de points.

Dans la suite de ce chapitre, nous commençons par présenter les algorithmes utilisant le critère du *minimax*. Dans une deuxième partie, nous présentons les algorithmes basés sur le critère du *maximin* qui utilisent une stratégie déterministe. Dans une troisième partie, nous présentons les algorithmes basés sur le critère du *maximin* qui utilisent une stratégie déterministe. Ces algorithmes ne convergeant pas vers la grille de Shukarev lorsque la taille de la suite et adéquat, nous présentons un nouvel algorithme dans la quatrième et dernière section.

Ce chapitre a fait l'objet, en plus des communications présentées au chapitre 3, de présentations à une conférence francophone (Gandar *et al.* (2010a)), à un workshop d'active learning d'une conférence internationale (Gandar *et al.* (2010b)) et à une conférence internationale (Gandar *et al.* (2011)).

Rappelons que la dispersion de la séquence $x = \{x_1, \ldots, x_n\}$ dans l'hypercube unité $I^s$ est définie par $\delta(x) = \sup_{y \in I^s} \min_{i=1,\ldots,n} d(y, x_i)$. La dispersion correspond au rayon de la plus grande boule vide de l'espace, et elle est bornée inférieurement par :

$$\delta(x) \geq \frac{1}{2\lfloor \sqrt[s]{n} \rfloor}. \tag{5.1}$$

## 5.1 Algorithmes basés sur le critère du *minimax*

Le critère du *minimax* correspond au critère de dispersion : il s'agit de la distance maximale entre tout point de l'espace et le point de la suite le plus proche. Mathématiquement, le critère du minimax s'écrit alors :

$$minimax(x) = \delta(x) = \sup_{y \in I^s} \min_{i=1,\ldots,n} d(y, x_i).$$

### 5.1.1 Algorithme d'échange de points

En se basant sur les travaux de Kennard et Stone (1969) et de Mitchell (1974) pour générer une suite « uniforme », Johnson *et al.* (1990) proposent de générer une suite à faible dispersion en utilisant un processus de *swapping*, *i.e.* d'échange de points. Le principe de base de cet algorithme consiste à déterminer un sous-ensemble de points, à partir d'un ensemble de candidats, qui améliore le critère de dispersion, c'est à dire la distance maximale entre les points du cube et les points de l'espace candidat.

Afin de trouver la configuration optimale au sens de la dispersion, l'algorithme utilise alors une distribution aléatoire uniforme et fait diminuer la dispersion à l'aide d'échange d'un ensemble fini de points candidats. Le principe de l'algorithme, présenté en 5.1.1, est le suivant : pour un point de la suite, on remplace ce point par un point de l'ensemble candidat (aléatoire ou non) et on examine si cet échange permet d'améliorer le critère. Si l'amélioration est avérée, le nouveau point est ajouté à la suite et l'ancien point est ajouté à l'ensemble des points candidats. Ce processus se termine lorsqu'il n'y a plus d'échanges possibles, qu'un nombre d'itérations préalablement fixé est atteint, ou que des permutations cycliques des points sont détectées.
La convergence de cet algorithme est garantie avec l'ensemble de points possibles, cependant nous n'avons aucune garantie d'avoir généré la suite à plus faible dispersion.

**Amélioration de l'algorithme :** l'estimation de la dispersion s'avère en pratique être très couteuse. Pour réduire les calculs, il est possible d'obtenir une évaluation de celle-ci en estimant pour chaque point de l'espace candidat, la distance minimale aux points de la suite et de prendre la distance maximale sur ces points candidats. La qualité de cette évaluation est cependant très dépendante du nombre et de la configuration des points de l'espace candidat. Royle et Nychka (1998) proposent un calcul efficace du critère de dispersion lorsque deux points de la suite sont échangés. Enfin, l'algorithme peut considérer uniquement les échanges entre les points candidats et les points de la suite les plus proches, ce qui permet de diminuer les temps de calcul lorsque l'ensemble de points candidats est grand.

**Algorithme 5.1.1** : Algorithme de génération d'une suite à faible dispersion selon le critère du *minimax* et par échange de points.

**Entrées** :
- suite à modifier $S$ ;
- ensemble $D$ de points candidats ;
- une fonction nommée *dispersion* d'évaluation de la dispersion de la suite $S$.

**répéter**
  **pour** $i$ =*1 à* $\#(S)$ **faire**
    $tempo = S(i)$ ;
    $d_0 = dispersion(S)$ ;
    **pour** $j$=*1 à* $\#(D)$ **faire**
      $S(i) = D(j)$ ;
      $d = dispersion(S)$ ;
      **si** $d_0 > dispersion(S)$
      **alors**
        break ;
      **sinon**
        $S(i) = tempo$ ;
        $j = j+1$ ;
      **fin**
    **fin**
  **fin**
**jusqu'à** *Critère d'arrêt* ;

**Sorties** : suite $S$ modifiée.

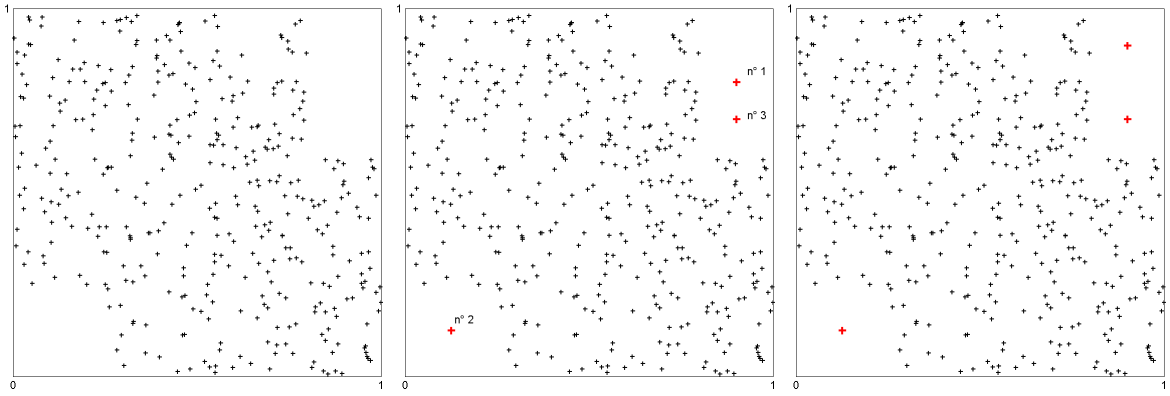
.

**Extension de l'algorithme :** une extension possible consiste à rajouter des points à une suite initialement fixée, *i.e.* à des points de la suite qui ne peuvent être échangés. Ceci est utilisé dans une approche séquentielle de génération de suite à faible dispersion : par exemple, après une première exploration d'un modèle, on souhaite re-générer des points dans une zone particulière de l'espace qui possède déjà des points issus de la première exploration. Cela suppose que l'exploration du modèle se fasse sur des critères géométriques de l'espace de ces paramètres et non sur une réponse qualitative du modèle.

### 5.1.2 Algorithme d'ajout de points

En se basant sur les travaux de LaValle et Branickly (2004), Lindemann et LaValle (2004) construisent un algorithme de réduction de dispersion en ajoutant des points pour des applications en robotique. Leur but est de construire des trajectoires d'un point initial à un point cible dans un espace rempli d'obstacles en faisant l'hypothèse qu'il existe au moins un chemin reliant les deux points et évitant les obstacles. Pour explorer ainsi l'espace, ils construisent itérativement une suite de points qui ne sont pas dans les obstacles. Ces points sont alors reliés et hiérarchisés entre eux par une structure d'arbre. A une itération donnée, ils rajoutent aléatoirement un point à la suite (ou une feuille à l'arbre). A chaque point est associée une cellule de Voronoï. L'ajout d'un point permet de réduire le volume d'une cellule de Voronoï. Le choix de la cellule à diviser est alors réalisé aléatoirement et proportionnellement au volume de-celle ci. Ainsi, en réduisant ces cellules, la dispersion de la suite est réduite itérativement.

**Avantage et inconvénient de cet algorithme :** cette méthode permet de réduire la dispersion globale de la suite tout en contournant le problème global de minimisation initiale. Cependant, ellen'est pas optimale lorsque le nombre de points maximal de la suite est fixé à l'avance. En effet, l'ajout de chaque point est optimal par rapport à la configuration précédente. Cependant, cette succession d'optimisations locales n'est pas globale. Afin d'illustrer cette non optimisation globale, nous avons représenté à la figure 5.1(a) une suite de 332 points en dimension 2 à laquelle nous souhaitons rajouter trois points supplémentaires. En ajoutant les points un par un, *i.e.* en faisant une succession d'optimisation locale, nous pouvons obtenir la configuration (ou une configuration équivalente) présentée à la figure 5.1(b) ; les numéros au-dessus des points représentent leur ordre d'ajout. Une manière plus optimale en terme d'uniformité de la distribution des points serait la distribution représentée à la figure 5.1(c) .




(a) Suite initiale de 332 points en dimension 2.

(b) Suite initiale à laquelle ont été rajoutés 3 points un par un. Les numéros au dessus des points indiquent l'ordre d'ajout de ceux-ci.

(c) Suite initiale à laquelle on été rajoutés 3 points de manière optimale.

**FIGURE 5.1:** *Illustration de la non optimalité globale de l'algorithme d'ajout de points un par un.*

### 5.1.3 Difficulté de générer des suites selon le critère *minimax*

Dans la littérature relative au *space filling design*, il est souvent admis que générer des plans à dispersion minimale est difficile. Yakowitz *et al.* (2000) disent que *« en dimension $s \geq 2$, pour chaque n, générer une suite de taille n dans l'hypercube unité qui minimise la dispersion est un problème difficile non-résolu, et la valeur optimale (de la dispersion) est inconnue »*. La génération

de tels plans dans le tore unité est plus aisée car le tore ne possède pas d'arêtes orientées et est résolue par les mêmes auteurs qui utilisent des lattices de Voronoï de premier type.

## 5.2 Algorithme basé sur le critère du *maximin*

La génération de suite à faible dispersion, ou selon le critère *minimax*, tout comme l'estimation de ce critère peut être être techniquement difficile : le nombre faible d'algorithmes générant de telles suites en utilisant ce critère en est un bon indicateur.
Johnson *et al.* (1990) proposent d'utiliser un autre critère pour générer des suites à faible dispersion : la critère dit du *maximin*. Ce critère ne prend plus en compte la distance entre les points de la suite et les points de l'espace comme la dispersion. Pour rappel, la dispersion de la suite $x$ est égale à :

$$\delta(x) = \sup_{y \in I^s} \min_{i=1,\ldots,n} d(y, x_i) \tag{5.2}$$

Le critère *maximin*, noté $\delta_2$ représente la distance minimale entre deux points de la suite et est défini par :

$$maximin(x) = \delta_2(x) = \inf_{(x_1,x_2) \in x \times x} d(x_1, x_2)$$

L'estimation de ce critère est beaucoup plus aisé et moins gourmand en nombre de calculs en général. Cependant, optimiser ce critère pousse en général les points hors de l'hypercube. Nous verrons comment les algorithmes limitent cet inconvénient.

### 5.2.1 Algorithme WSP de suppression

En s'inspirant des travaux initiaux en chimie de Wooton *et al.* (1975) et en space filling design de Kennard et Stone (1969), Sergent *et al.* (1997) définissent l'algorithme WSP qui se base sur la sup pression de points à une suite initiale pour obtenir une suite dont la dispersion voulue ait la valeur $\delta$. L'algorithme présenté en 5.2.1 est le suivant : initialement, l'algorithme dispose dans l'hypercube unité une suite aléatoire uniforme de très grande taille. L'algorithme sélectionne alors le point $\hat{x}$ de la suite le plus proche du centre de l'hypercube. Ensuite, l'algorithme élimine de la suite les points situés dans la boule de centre $\hat{x}$ et de rayon $\delta$, excepté $\hat{x}$. L'algorithme réitère ce processus en considérant une nouvelle sphère d'élimination centrée au point le plus proche du centre de la sphère précédente. Lorsque tous les points de la distribution initiale sont parcourus (éliminés ou conservés), le processus est terminé.
L'algorithme délivre alors une suite dont la taille est inférieure à la taille de la suite initiale et dont la dispersion est inférieure ou égale à $\delta$ si la suite initiale est de taille suffisamment grande et si elle recouvre bien « uniformément » tout l'espace. L'algorithme assure que chaque point de la suite est bien espacé d'une distance au moins de $\delta$ de ses voisins, et aussi près que possible du centre de l'hypercube.

**Inconvénients de l'algorithme :** le principal inconvénient réside dans la difficulté de prédire le nombre de points souhaité de la suite finale, ainsi que le choix de la distance minimale pour s'en approcher. De plus, les défauts de la distribution initiale resteront dans la suite finale : si des zones de l'espace sont vides avec la distribution initiale, ils le resteront dans la suite finale. Pour amoindrir ces phénomènes, il est préférable de générer une distribution initiale de grande taille et déjà bien répartie, comme par exemple une suite à discrépance faible. De plus, cette dernière possibilité permet d'obtenir une suite dont la discrépance sera en général plus faible que celle d'une suite aléatoire.

**Algorithme 5.2.1** : Algorithme selon le critère du *maximin* de suppression de points pour générer une suite de dispersion fixée $\delta$.

1. Générer une suite aléatoire uniforme $X = \{x_1, \ldots, x_N\}$ dans $I^s$ de taille $N >> n$ excessivement grande ;
2. trouver le point $x^* \in X$ le plus proche du centre $(1/2, \ldots, 1/2)$ de l'hypercube ;
3. trouver l'ensemble $\tilde{E}$ des points de $E$ situés à une distance de $x^*$ inférieure à $\delta$ ;
4. $E = E - \tilde{E}$,

**répéter**

1. trouver le point $x^{**}$ de $E$ le plus proche de $x^* \in X$ :
2. $x^* = x^{**}$ ;
3. trouver l'ensemble $\tilde{E}$ des points de $E$ situés à une distance de $x^*$ inférieure à $\delta$ ;
4. $E = E - \tilde{E}$,

**jusqu'à** *Tous les points de la suite initiale E ont été parcourus (soit éliminés, soit conservés)* ;

**Sorties** : suite $S$ modifiée.

### 5.2.2 Algorithme d'ajout de points

Une autre type d'algorithme consiste à ajouter des points un par un à une séquence initiale en réduisant la dispersion. Cette technique est commune aux algorithmes utilisant les deux critères. Un de ces algorithmes a été présenté au paragraphe 5.1.2 avec le critère *minimax*. Celui-ci étant couteux à estimer, il peut être remplacé par le critère de *maximin*. Cependant, les algorithmes basés sur ce critère maximisation de $\delta_2(x) = \inf_{(x_1,x_2)\in x\times x} d(x_1,x_2)$ poussent généralement les points vers les frontières de l'hypercube. Cet effet peut être corrigé en introduisant une interaction entre les points de la suite et les frontières du cube. Pour cela, au lieu d'utiliser le critère $\delta_2$, nous pouvons utiliser le critère incluant l'interaction suivante :

$$\delta_3(x) = \inf_{(x_1,x_2)\in x\times x} d\left(x_1, \{x_2\}\cup\{I^s\}'\right) \text{ où } \{I^s\}' = \{x\in\mathbb{R}^s | x\notin I^s\}. \tag{5.3}$$

En s'inspirant des travaux de Lindemann et LaValle (2004), Teytaud *et al.* (2007) ont développé un algorithme utilisant ce critère $\delta_3$ et utilisant des arbres aléatoires pour explorer l'espace et choisir où placer le nouveau point. A la première itération, l'algorithme génère un point au milieu du centre de l'hypercube, puis, à chaque itération, l'algorithme ajoute un point en minimisant le critère $\delta_3$.

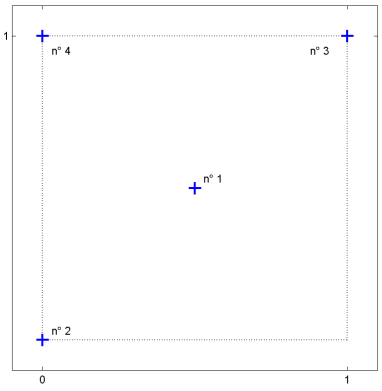
**Inconvénient de cet algorithme :** comme pour l'algorithme d'ajout de points à une suite en utilisant le critère du *minimax* à la section 5.1.2, l'ajout d'un point est réalisé de manière optimale par rapport à la configuration de l'étape précédente : le critère est alors optimisé. Cependant, cette optimisation est locale et ne devient pas globale avec les itérations.
Cela se remarque facilement avec la figure 5.2 : nous avons représenté en dimension 2 aux figures 5.2(a) & 5.2(c) les suites de taille 4 et 9 obtenues avec cet algorithme. Nous les avons comparées avec les grilles de Shukarev de mêmes tailles aux figures 5.2(b) & 5.2(d). Nous pouvons remarquer que les configurations obtenues avec l'algorithme ne sont pas optimales. Cette non-optimalité est illustrée graphiquement avec des tailles correspondant aux grilles de Shukarev. Nous obtenons le même résultat pour des tailles quelconques : nous pourrions trouver des configurations ayant des dispersions inférieures à celles obtenues par cet algorithme en utilisant par exemple l'algorithme de Gandar *et al.* (2011).

## 5.3 Génération stochastique des plans *maximin*

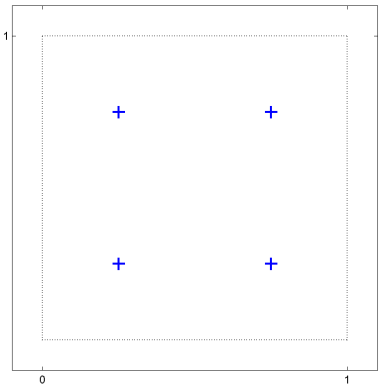
La fonction $\delta : (I^s)^n \to \mathbb{R}^+$ qui, à un ensemble de points $E = \{x_1,\ldots,x_n\}$ associe la dispersion, est une fonction continue définie sur le compact $(I^s)^n$ ; par conséquent elle admet une configuration des points qui la minimise. Néanmoins, dans la majorité des cas, cette configuration n'est pas nécessairement unique : seul le cas où la taille $n$ de l'ensemble $E$ est adéquate admet comme unique solution la grille de Shukarev. Dans le cas où la taille n'est pas adéquate, il n'existe aucune solution théorique. Différents algorithmes stochastiques ont été créés afin d'estimer une solution de ce problème. Nous présentons dans un premier temps un algorithme stochastique de recuit-simulé, puis dans un deuxième temps, nous présentons un algorithme basé sur le processus de Strauss.

### 5.3.1 Algorithme de recuit simulé

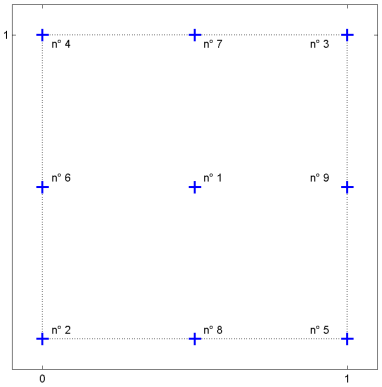
Partant du principe que le domaine de définition d'un plan d'expérience n'est pas nécessairement hypercubique, Auffray *et al.* (2012) ont récemment proposé un algorithme de génération de points selon le critère du *maximin* dans tout ensemble borné $D$. Celui-ci est un algorithme de recuit simulé. Le principe de cet algorithme est de forcer les paires de points qui sont trop proches à être plus espacés.



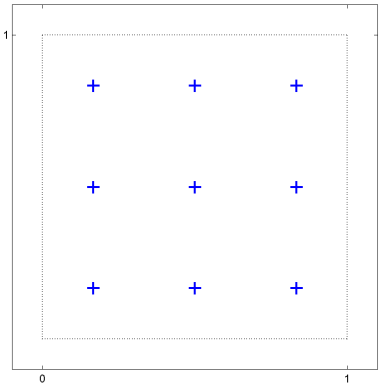

(a) 4 points générés selon l'algorithme proposé par Teytaud *et al.* (2007), dispersion : 0.71.




(b) Grille de Shukarev de 4 points, dispersion : 0.36.




(c) 9 points générés selon l'algorithme proposé par Teytaud *et al.* (2007), dispersion : 0.35.




(d) Grille de Shukarev de 9 points, dispersion : 0.24.

**FIGURE 5.2:** *Illustration en dimension 2 de la non optimalité globale de l'algorithme de Teytaud et al. (2007) d'ajout de points un par un sur des suites de taille 4 et 9. Sur les figures 5.2(a) & 5.2(b), les nombres indiquent l'ordre dans lequel les points ont été ajoutés un par un. Les grilles de Shukarev 5.2(c) & 5.2(d) sont les meilleures configurations pour ces nombres de points adéquats.*

La procédure d'initialisation consiste à générer aléatoirement un très grand nombre de points, à estimer leur covariance empirique $\hat{\sum}$ et enfin à sélectionner parmi ces points les $n$ points qui formeront un premier plan d'expériences, noté $X^{(0)} = \left\{x_1^{(0)}, \ldots, x_n^{(0)}\right\}$.

Les procédures itératives sont ensuite expliquées à l'algorithme 5.3.1. Dans cet algorithme, la température $T_t$ et la variance $\sigma_t$ décroissent avec le temps avec une décroissance inversement logarithmique. La fonction $q$ désigne un noyau de transition entre les configurations. Ce noyau s'exprime avec la fonction de répartition de la loi normale utilisée, et la fonction $\delta$ est définie ainsi $\delta_X = \min_{i,j\in\{1,\ldots,n\}} \left\|x_i - x_j\right\|^2$.

---

**Algorithme 5.3.1** : Algorithme de type *maximin* de recuit-simulé de simulations de points à faible dispersion de Auffray *et al.* (2012).

---

**Entrées** :
- ensemble initial de $n$ points $X^{(0)} = \left\{x_1^{(0)}, \ldots, x_n^{(0)}\right\}$ ;
- matrice de variance-covariance $\hat{\sum}$ de $X^{(0)}$.

1. Choisir une paire de points $\left(x_i^{(t)}, x_j^{(t)}\right)$ dans $X^{(t)}$ suivant une loi multinomiale dont les probabilités sont proportionnelles à $\dfrac{1}{\left\|x_i^{(t)} - x_j^{(t)}\right\|}$ ;
2. choix avec la probabilité $\dfrac{1}{2}$ de l'un des deux points, noté $x_k^{(t)}$ ;
3. utilisation d'une marche aléatoire gaussienne pour proposer un nouveau point :

$$x_k^{prop} \sim \mathcal{N}_s\left(x_k^{(t)}, \sigma_t \hat{\sum}\right),$$

   la configuration proposée est alors notée :
   $X^{prop} = \left\{x_1^{(t)}, \ldots, x_{k-1}^{(t)}, x_k^{prop}, x_{k+1}^{(t)}, \ldots, x_n^{(t)}\right\}$ ;
4. si $X_k^{prop} \in D$, accepter $X^{(t+1)} = X^{prop}$ avec la probabilité

$$\min\left\{1, \frac{q\left(X^{(t)}|X^{prop}\right)}{q\left(X^{prop}|X^{(t)}\right)} \exp\left(\frac{\delta_{X^{prop}} - \delta_{X^{(t)}}}{T_t}\right)\right\},$$

   sinon $X^{(t+1)} = X^{(t)}$.

**Sorties** : ensemble initial modifié de $n$ points.

---

Cet algorithme permet de générer une suite à faible dispersion quel que soit l'espace, ce qui en fait un grand avantage. Ceci est possible par l'utilisation du critère du *maximin*. Cet algorithme prend en compte également le volume dans lequel nous souhaitons générer des points d'apprentissage par le biais du choix du paramètre initial $\sigma_0$ d'écart-type de la loi normale utilisée pour

simuler un nouveau point. Les auteurs préconisent d'utiliser la valeur $\frac{\lambda(E)}{n^{-1/s}}$, et d'utiliser la descente suivante avec les itérations $\sigma_n = \frac{\sigma_0}{\sqrt{n}}$. Enfin, aucun critère d'arrêt n'est fourni : la convergence de l'algorithme est théoriquement garantie à l'infini. En pratique, les auteurs se contentent d'itérer l'algorithme un grand nombre de fois, *e.g.* $10^7$ fois pour générer un plan d'expérience de 2 249 points en dimension 8. Cet algorithme fournit de bonnes dispositions de points dans les espaces quelconques. Néanmoins, lorsque l'espace est un hypercube et que le nombre de points est adéquat, la disposition délivrée n'est pas une grille de Shukarev.

### 5.3.2 Algorithme du processus de Strauss

Les processus de Strauss considèrent les $n$ points comme étant $n$ particules chargées électriquement qui se repoussent entre elles. Pour générer de tels processus, Franco (2008) utilise des techniques de simulation par chaînes de Markov et l'algorithme de Metropolis-Hastings en se basant sur les travaux de Ripley et Kelly (1977). La loi de processus conditionnée par le nombre de points $n$ est donnée par :

$$\pi(X) = k * \gamma^{S(X)}$$

où $0 < \gamma < 1$ est un coefficient de répulsion, $k$ une constante de normalisation, et la fonction $s(x)$ représente un potentiel global d'énergie. Cette fonction est définie par :

$$s(X) = s(x_1, \ldots, x_n) = \sum_{i<j} \mathbb{1}_{\{d(x_i,x_j)<R\}}.$$

Concrètement, il s'agit du nombre de couples de points $(x_i, x_j)$ tels que la distance entre les points $x_i$ et $x_j$ soit inférieure ou égale à $R$. En termes de particules chargées, cela représente le nombre global d'interactions limitées uniquement aux interactions de paires (dites d'ordre 2). Chaque particule a alors une sphère d'influence de rayon $R/2$ centrée en elle, l'interaction se produisant exactement lorsque 2 sphères quelconques se rencontrent. Le choix de la valeur du paramètre $R$ est une question cependant délicate de l'algorithme. En effet, aucune règle de choix n'est délivrée. Une valeur petite du paramètre $R$ ne permet pas de bien répartir les points sur tout le domaine, et la configuration obtenue pour les points est proche de celle d'une suite aléatoire uniforme. A l'inverse, une valeur de $R$ trop grande donne une zone d'influence très grande pour chaque point, et l'algorithme reste dans une configuration optimale locale et non globale. Cela se caractérise par la présence de clusters de points : dans un même cluster, les points sont proches, et les centroïdes des clusters sont éloignés. Dans sa thèse, l'auteur considère que ce rayon d'interaction est le paramètre le plus sensible à régler, et que, pour un critère donné, la meilleure solution serait sans doute de tabuler cette valeur selon le nombre de points et la dimension du problème. Nous pouvons donc remarquer que le critère de dispersion défini par Niederreiter (1992) n'intervient nullement ici.

Le paramètre $\gamma$ permet de d'obtenir une probabilité plus ou moins forte aux plans pour lesquels les expériences sont reparties de manière à interagir plus ou moins les unes avec les autres. Le plan est d'autant plus de bonne qualité que le nombre d'interactions $s(x)$ est faible. Le cas où $\gamma = 0$ interdit toute paire de points de distance inférieure à $R$, et conduit à des points régulièrement espacés. Cependant le choix du paramètre $R$ est difficile. Le cas où $\gamma = 1$ correspond exactement à l'indépendance des points, *i.e.* à une distribution aléatoire uniforme.

L'estimation des valeurs adéquates des paramètres $\gamma$, et $R$ est néanmoins difficile comme nous avons pu l'énoncer avant. De plus, il n'existe pas d'expression analytique explicite pour le choix de la valeur de la constante de normalisation $k$, même pour des tailles modestes de $n$ et de $s$. Pour cela, Franco (2008) utilise des méthodes de Monte-Carlo par chaîne de Markov (MCMC) pour estimer le paramètre $k$ et couple cette méthode avec l'algorithme de Metropolis-Hastings (MH). Dans le contexte de génération de plans d'expériences, un état de la chaîne de Markov est une suite de $n$ points.

L'algorithme de MH, présenté en 5.3.2, va procéder à chaque itération en 2 étapes distinctes :

1. une phase de proposition de changement d'état (plan d'expérience) où le nouvel état ne diffère que par une seule expérience (un seul point) ;
2. une autre, d'acceptation ou de rejet de ce changement.

---

**Algorithme 5.3.2** : Algorithme selon le critère du *maximin*de MH pour générer des suites à faible dispersion par processus de Strauss et MCMC.

---

**Entrées** : suite aléatoire uniforme $\{x_1,\ldots,x_n\}$.

**pour** *i=1 à n* **faire**
  **pour** *j=1 à n* **faire**
    – choisir une expérience $x_i$ au hasard ;
    – simuler $y_i$ uniformément dans $I^s$ ;
    – accepter le changement $x_i = y_i$ avec la probabilité :
      $a(X,Y) = \min\left(1, \frac{\pi(Y)}{\pi(X)}\right)$.
      où $Y = (x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n)$.
  **fin**
**fin**

**Sorties** : suite modifiée $\{x_1,\ldots,x_n\}$.

---

La grande force de cet algorithme tient au fait que le calcul du rapport $\frac{\pi(Y)}{\pi(X)}$ revient à chaque fois à calculer le nombre $s(x_i) = \sum_{j\neq i} \mathbb{1}_{\{d(x_i,x_j)<R\}}$ de voisins du point $x_i$, *i.e.* les points contenus dans la sphère centrée en $x_i$ et de rayon $R$, de façon identique et le nombre $s(y_i) = \sum_{j\neq i} \mathbb{1}_{\{d(y_i,x_j)<R\}}$ de voisins du point $y_i$, *i.e.* les points contenus dans la sphère centrée en $y_i$ et de rayon $R$. On peut vérifier alors que $\frac{\pi(Y)}{\pi(X)} = \gamma^{s(y_i)-s(x_i)}$.

Afin de limiter les agrégats de points, notamment lorsque le rayon est inadapté, il est possible de généraliser cet algorithme en modifiant l'estimation des forces d'interactions entre points. Par exemple, ces interactions peuvent être mesurées par la distance réelle entre les points et non plus par la seule connaissance d'interaction ou pas (sphères de rayon $R/2$ qui se rencontrent). Il s'agit alors du processus de Strauss-Gibbs.

Cet algorithme a la particularité d'être peu gourmand en calculs, néanmoins il se caractérise par de nombreux paramètres, qui, en pratique, s'avèrent être difficiles à estimer.

Le principe de cet algorithme peut être généralisé à d'autres problématiques de génération de points. En supposant que l'on ait des connaissances particulières sur le phénomène étudié, on peut alors souhaiter ne plus échantillonner selon le critère de dispersion minimale, mais selon une distribution plus dense dans une certaine partie de l'hypercube. Ainsi, il doit être possible de générer des points selon une distribution non plus uniforme mais hétérogène ou bien encore anisotrope. Cette généralisation se réalise en prenant d'autres définitions de la densité $\pi$, en prenant par exemple des densités de lois gaussiennes.

## 5.4 Algorithme de recuit-simulé selon le critère du *minimax* prenant en compte le critère de la dispersion (RSCM)

Tous les algorithmes présentés à la section précédente sont stochastiques et délivrent des solutions proches des solutions optimales. Ils permettent d'établir différentes suites à dispersion équivalente, ce qui peut être utile pour étudier des phénomènes stochastiques. Leurs avantages résident principalement dans le fait qu'ils sont peu gourmands en calculs et peuvent donc être mis en place facilement. Des autres avantages importants résident dans le fait qu'ils peuvent générer des suites de points selon des critères autres que l'« uniformité » pour l'un, et dans le fait qu'ils peuvent s'appliquer dans des espaces différents de l'hypercube unitaire pour l'autre. Néanmoins, ces algorithmes ne convergent pas nécessairement vers une solution optimale. En effet, lorsque le nombre de points est adéquat, la configuration délivrée par les algorithmes n'est en général pas une grille de Shukarev.

Nous proposons alors un nouvel algorithme, le RSCM *i.e.* **Recuit Simulé selon le Critère du *Minimax***, pour générer une suite à faible dispersion en utilisant :

1. la propriété 5.1 sur la borne inférieure de la dispersion d'une suite,
   *i.e.* $\delta(x) \geq \frac{1}{2\lfloor\sqrt[s]{n}\rfloor}$ ;
2. une approche de type *maximin* ;
3. des variables ressorts qui déplacent les points en fonction de leurs distances entre eux et de leurs distances aux frontières de l'hypercube, comme les processus de Strauss.

Bien qu'initialement l'algorithme RSCM ne fut pas créé pour ses propriétés calculatoires et algorithmiques, il possède en pratique de bonnes propriétés, et a donc toute sa place dans ce mémoire de thèse. Lorsque le nombre de points de la suite est approprié, l'algorithme converge vers la grille de Shukarev qui est la configuration optimale. A notre connaissance, aucun algorithme ne réalise cette convergence. De plus, celle-ci est assez rapide. La complexité de RSCM est de l'ordre de $\mathcal{O}(n^2k)$ pour une suite de $n$ points après $k$ itérations. Enfin, d'après notre connaissance, cet algorithme est le premier à considérer la propriété 5.1 sur la borne inférieure de la dispersion d'une suite.

Dans la suite de cette section, nous décrivons cet algorithme RSCM, ainsi qu'une extension.

### 5.4.1 Description de l'algorithme RSCM

L'algorithme, écrit en pseudo-code à l'algorithme 5.4.1 et illustré à la figure 5.3, s'appuie sur les étapes suivantes :

**Initialisation :** nous appliquons l'algorithme à une suite initialement existante, sinon nous générons aléatoirement et uniformément une première suite $S$ à $n$ points, préférablement au milieu de l'hypercube $S = \{x_i\}_{i=1,\dots,n}$ en dimension $s$. A chaque étape, chaque point repousse ses voisins qui sont à une distance inférieure à $d_m = \frac{1}{\lfloor\sqrt[s]{n}\rfloor}$. En effet, grâce à l'inégalité 4.2, nous avons la relation $\delta(S) \geq \frac{1}{2}d_m$.

**Mouvement des points :** pour chaque point $x$ de $S$, nous ne considérons comme voisins de $x$, que les points $x_i$ de $S$ situés à une distance inférieure à $d_m$. Nous calculons alors une variable ressort entre le point $x$ et chacun de ses voisins. Cette variable est définie par $\left(\frac{2*d_m - d(x,x_i)}{2*d_m}\right)^p$. Le paramètre $p$ est un entier positif, et nous avons observé expérimentalement que la valeur $p = 4$ est satisfaisante. Avec $p = 4$, la valeur de la variable ressort varie entre $1/16$ (pour les points les plus

loins de x) et 1 (pour les points les plus proches). Ensuite, nous bougeons le point $x$ proportionnellement à chaque variable ressort selon la direction du vecteur $x - x_i$ : plus les points sont proches les uns des autres, plus l'algorithme les écarte. Enfin, la proportionnalité d'écartement est décroissante avec le temps selon le principe du recuit-simulé, jusqu'à un temps seuil à partir duquel elle devient constante. Cette étape de l'algorithme est présentée aux figures 5.3(a) et 5.3(b).

Ce processus est similaire à l'approche *minimax* et pousse donc les points à l'extérieur de l'hypercube ou sur les arrêtes de celui-ci. Une illustration est présentée aux figures 5.3(c) et 5.3(d). Le problème de la génération de suites à faible dispersion consiste alors en la minimisation d'un critère sous contraintes de boîtes. Ainsi, nous ajoutons ces contraintes de cette sorte : les bords du cube soumettent également aux points une force répulsive en direction du centre du cube, dont l'intensité dépend de la distance entre les points et les bords du cube.

**Applications de contraintes de boîtes :** de manière à garder les points à l'intérieur de l'hypercube, nous appliquons alors des forces répulsives, présentées ci-dessus, aux points les plus proches du bord. Ces points sont détectés par une de leurs coordonnées qui est, soit inférieure à $\frac{d_m}{2} + \varepsilon_m$, soit supérieure à $1 - \frac{d_m}{2} - \varepsilon_m$ : ces valeurs correspondent aux coordonnées extremum des points d'une grille de Shukarev [1] avec une tolérance de $\varepsilon_m = \frac{d_m}{4}$. L'intensité de la force possède les mêmes propriétés que celles utilisées pour l'interaction entre les points de la suite. Cette étape de l'algorithme est présentée à la figure 5.3(e).

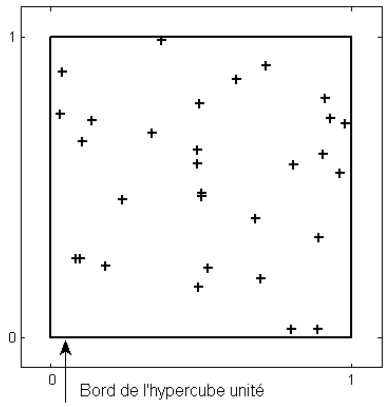
Globalement, nous réalisons ainsi une minimisation locale de la dispersion, qui devient globale après plusieurs itérations de ce processus ...

**Limiter les configurations à minimum local :** en réalisant itérativement les deux étapes précédentes, l'algorithme peut alors tomber dans une configuration de minimum local, où les points oscillent sur eux-mêmes : un nombre élevé de points se retrouvent alignés sur une (ou des) arrête(s) de l'hypercube. Les forces répulsives poussent les points entre eux, ayant tendance ainsi à les pousser vers l'extérieur du cube. Les forces répulsives des bords ont tendance, elles, à les ramener à l'intérieur. Ces deux types de forces s'annulent alors et provoquent ainsi des oscillations des points. De manière à éviter le développement de ces configurations à minimums locaux, après quelques itérations de l'algorithme, nous choisissons aléatoirement un point sur chaque bord de l'hypercube et changeons sa coordonnée selon une variable aléatoire uniforme centrée sur le milieu de l'hypercube.

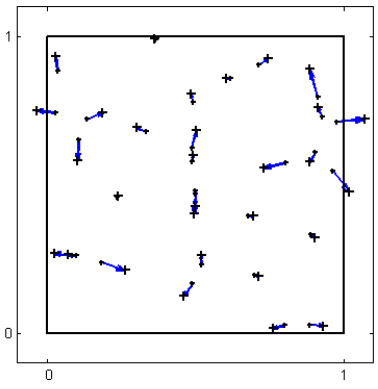
**Critère d'arrêt :** différents critères d'arrêt peuvent être utilisés, tels qu'un nombre maximal d'itérations de l'algorithme, la stabilisation du critère de dispersion si on peut l'évaluer, ou bien alors l'atteinte d'un seuil minimum de la moyenne des normes des vecteurs de changement des points à une itération.

**Inconvénients de cet algorithme :** cet algorithme possède une convergence rapide en nombre d'itérations, mais nécessite cependant des capacités de stockage et des ressources de calcul importantes. En effet, il s'avère nécessaire de calculer les distances entre tous les points de la suite, ce qui peut demander un temps de calcul élevé lorsqu'il y a beaucoup de points et que cela est fait itérativement, ou alors, lorsque ces calculs sont réalisés vectoriellement, cela nécessite d'avoir des capacités de calculs importantes.
Une manière de s'affranchir de ce problème consiste à ne pas calculer toutes les distances entre

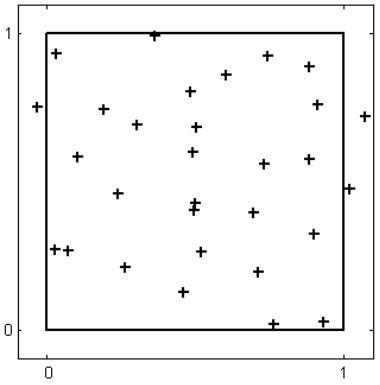
1. Grille de taille adéquate directement inférieure à la taille de la suite.



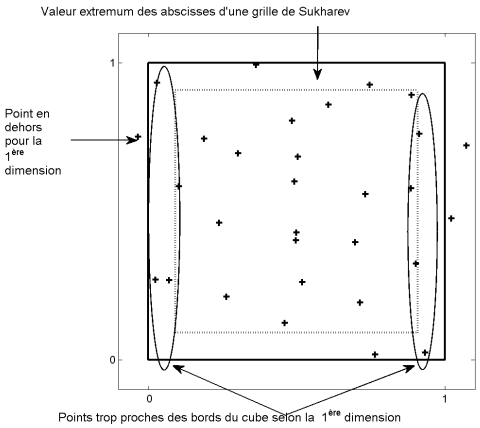

(a) Suite initiale de 30 points dont on souhaite réduire la dispersion.



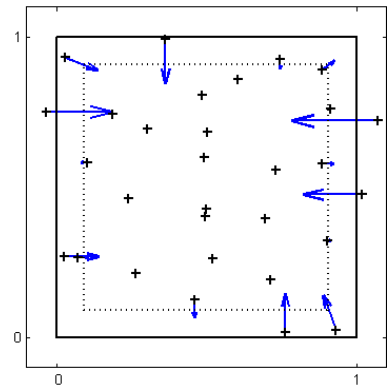

(b) Processus de répulsion des points entre eux. Les points initiaux sont représentés par des points, les forces s'exerçant sur eux par des flèches de longueur proportionnelle à l'intensité, et les points déplacés sont représentés par des grosses croix.



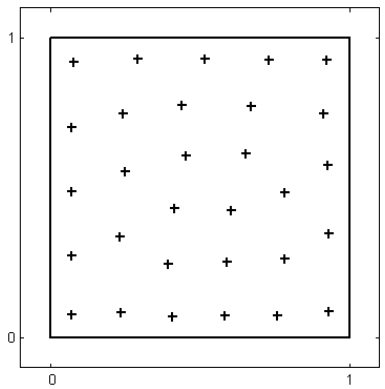

(c) Suite obtenue après le déplacement des points par répulsion entre eux.




(d) Ce processus repousse les points vers l'extérieur du cube ou vers ses arrêtes.




(e) Processus de répulsions des points trop proches des arrêtes du cube.




(f) Suite obtenue après plusieurs itérations de l'algorithme. On peut remarquer un positionnement régulier des points.

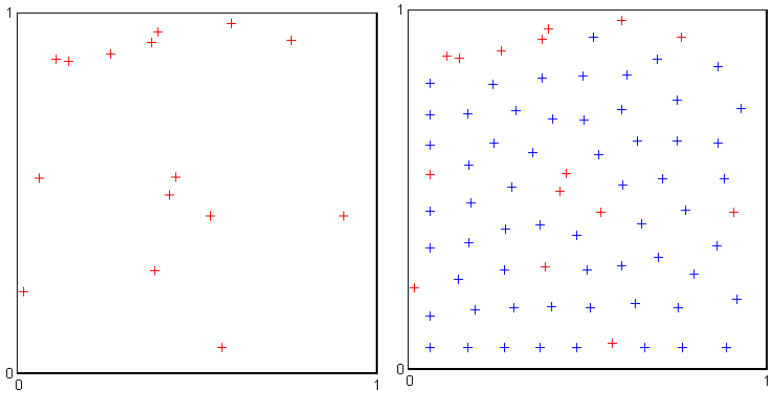
**FIGURE 5.3:** *Illustration en dimension 2 de l'algorithme RSCM de réduction de la dispersion défini par Gandar et al. (2011).*

tous les points. En effet, un point très éloigné d'un autre point n'influence pas ce dernier. Par conséquent, il n'est pas nécessaire de calculer toutes les distances, mais seulement celles pour des points qui sont dans des zones proches de l'espace. La méthode employée pourrait être celle présentée pour l'estimation de la dispersion au paragraphe 4.1 à la page 62.

### 5.4.2 Extension de l'algorithme

Dans cette partie, nous faisons une digression sur la problématique de génération initiale de $n$ premiers points à faible dispersion. Nous sortons de ce contexte de première génération de points d'exploration. Supposons que nous ayons une première suite de $n$ points, avec une certaine dispersion, et que nous souhaitions rajouter $n'$ autres points. Il est possible d'ajouter ces $n'$ autres points de manière itérative en utilisant les algorithmes d'ajout de points présentés aux sections 5.1.2 et 5.2.2. Comme présenté à la figure 5.1, l'ajout itératif de ces points étape par étape est optimal. Cependant cette succession d'optimisations locales ne délivre pas une optimisation globale.

L'algorithme RSCM présenté à la section précédente peut être étendu de manière à rajouter de nouveaux points, selon le critère de minimisation de la dispersion, à une suite déjà existante. Pour cela, il suffit de générer aléatoirement uniformément les $n'$ points dans le cube unité. Puis, on applique l'algorithme RSCM en ne bougeant que les points générés, les points de la suite initiale restant immobiles. Ce nouvel algorithme fournit en général de bons résultats comme le présente expérimentalement la figure 5.4.




(a) Suite initiale de 30 points à laquelle on souhaite ajouter des points.

(b) Suite initiale en rouge à laquelle on a ajouté 60 points bleus de manière à minimiser la dispersion en utilisant l'algorithme de Gandar *et al.* (2011).

**FIGURE 5.4:** *Illustration en dimension 2 de l'algorithme d'ajout de 60 points RSCM à une suite initiale de 15 points selon le critère de la dispersion en utilisant l'algorithme RSCM.*

**Algorithme 5.4.1** : Algorithme de recuit-simulé selon le critère du minimax prenant en compte le critère de la dispersion définit par Gandar *et al.* (2011).

**Entrées** : suite à modifier $S = \{x_1, \ldots, x_n\}$ ;

$S2 \leftarrow S$ ; compteur $\leftarrow 1$ ; $d_m \leftarrow \frac{1}{\lfloor \sqrt[s]{n} \rfloor}$ ; $\varepsilon_m \leftarrow \frac{d_m}{4}$

**répéter**
- (Mouvement des points)
- **pour** *tout* $x \in S$ **faire**
  - move $\leftarrow 0$
  - **pour** *tout* $y \in S | y \neq x$ **faire**
    - **si** $d(x,y) < d_m$ **alors**
      - $t = \left(\frac{2*d_m - d(x,y)}{2*d_m}\right)^p$
      - $\text{move} = \text{move} + \mu * \left(0.55 + 0.45 * \max\left(1 - \frac{\text{compteur}}{\text{compteur}_{\max}}, 0\right)\right) * (x - y)$
    - **fin**
    - $S2(x) \leftarrow S(x) + \text{move}$
  - **fin**
- **fin**
- (Applications de contraintes de boîtes)
- **pour** *tout* $x \in S2$ **faire**
  - **pour** $i = 1$ *à* $s$ **faire**
    - **si** $x^i < \frac{d_m}{2} + \varepsilon_m$ **alors**
      - $x^i = \frac{d_m}{2} - S2(x)^i$
    - **fin**
    - **sinon si** $x^i > 1 - \frac{d_m}{2} - \varepsilon_m$ **alors**
      - $x^i = 1 - \frac{d_m}{2} - S2(x)^i$
    - **fin**
    - **sinon**
      - $x(i) = 0$
    - **fin**
  - **fin**
- **fin**
- $S(x) \leftarrow S(x) + \mu - 2 * \text{move} * \left(0.55 + 0.45 * \max\left(1 - \frac{\text{compteur}}{\text{compteur}_{\max}}, 0\right)\right)$
- (Limiter les configurations avec minimums locaux)
- **si** *compteur* $\in \{$*valeurs critiques*$\}$ **alors**
  - **pour** $i = 1$ *à* $s$ **faire**
    - Trouver $A = \left\{x \in S | x^i < \frac{d_m}{2} + \varepsilon_m\right\}$
      $a = randn(\#(A))$
      $S(A(a)) = rand(1,1) * 0.5 + 0.1$
    - Trouver $A = \left\{x \in S | x^i > 1 - \frac{d_m}{2} + \varepsilon_m\right\}$
      $a = randn(\#(A))$
      $S(A(a)) = rand(1,1) * 0.5 + 0.1$
  - **fin**
- **fin**
- compteur ++ ;

**jusqu'à** *Critère d'arrêt ou compteur* $>$ *compteur*$_{max}$ ;

**Sorties** : suite $S$ modifiée.

.

## 5.5 Dispersion et augmentation de la dimension

Dans cette section, nous cherchons à qualifier le comportement de la dispersion d'une suite aléatoire en fonction de la dimension. Comme il n'existe pas de résultats théoriques, nous présentons des résultats expérimentaux et nous nous intéressons alors au comportement de la dispersion des suites aléatoires uniformes par rapport à des configurations de points « minimisant » la dispersion. Pour cela, nous réalisons l'étude dans le cas où le nombre de points est tel que la configuration minimisant la dispersion est connue, *i.e.* la grille de Shukarev.

Expérimentalement, nous nous sommes placés dans le cas de suites à 2 points par dimension en dimension 2 à 12. Pour chaque dimension, nous avons généré 20 suites aléatoires uniformes et nous avons estimé leur dispersion. Les résultats obtenus sont présentés à la figure 5.5.

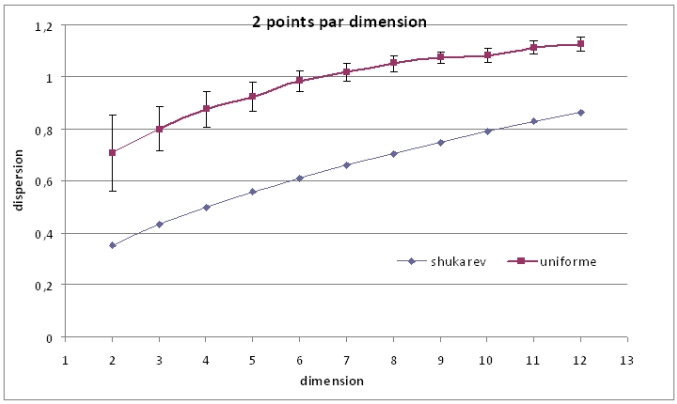



**FIGURE 5.5:** *Comparaison de la dispersion moyenne de suites aléatoires par rapport à la dispersion de la grille de Shukarev en fonction de la dimension de l'espace. Moyenne obtenue sur 20 suites aléatoires. Les intervalles de confiance sont donnés à un écart-type.*

**Interprétation :** nous pouvons remarquer que les suites aléatoires ont une dispersion moyenne significativement supérieure à la dispersion des grilles de Shukarev.
Tout comme les suites à faible discrépance (voir la section 2.3 à la page 42), les grilles de Shukarev souffrent cependant aussi de ce fléau : pour obtenir une dispersion d'une certaine qualité, il faut un nombre de points qui croît exponentiellement lorsque la dimension augmente.
Néanmoins, à la différence avec le critère de discrépance, lorsque la dimension augmente, le comportement des suites aléatoires reste significativement différent de celui des suites minimisant le critère (ici de dispersion). Cette différence est très importante : pour la discrépance, lorsque la dimension augmente, les comportements deviennent équivalents très rapidement (dès la dimension 6 ou 7) entre suites à discrépance faible et suites aléatoires, et il est alors moins coûteux de générer une suite aléatoire qu'une suite à faible discrépance (minimisant la discrépance). Dans le cas de la dispersion, le comportement entre suites aléatoires et grilles de Shukarev reste significative même pour des dimensions supérieures à 7. Ainsi, lorsque la dimension augmente, pour un nombre de points adéquat et en terme de dispersion, il est préférable d'utiliser des grilles plutôt que des suites aléatoires.
Il est probable que ce résultat s'étende (mais il faudrait le tester expérimentalement) au cas des suites de taille quelconque : l'utilisation de suites minimisant la dispersion serait alors préférable aux suites aléatoires.

## 5.6 Conclusion

La notion de dispersion, que nous avons présentée au chapitre 4, est une notion ancienne introduite par Niederreiter dans les années 1980. De nombreux problèmes scientifiques se résolvent par l'utilisation d'une suite à faible dispersion : la génération des premiers points d'apprentissages en classification active en est un bon exemple. La problématique de générer, pour une taille quelconque, une suite à faible dispersion, est donc très importante.

Johnson *et al.* distinguent alors dans les années 1990 deux méthodes, *minimax* et *maximin*, pour générer des suites minimisant la dispersion. Des premiers algorithmes ont été créés permettant la résolution de ces problèmes de génération de ces plans d'expériences à faible dispersion. Ces algorithmes furent surtout créés pour résoudre des problèmes faisant appel à la minimisation de la dispersion, mais cette minimisation n'était pas au cœur de ces travaux. Par conséquent, les algorithmes proposés n'étaient pas optimaux. Ce problème de génération de tels plans à faible dispersion a connu un renouveau depuis la fin de la première décennie des années 2000, grâce notamment aux progrès informatiques et à l'essor de la théorie des « computer experiments », ce qui se traduit par l'élaboration de nouveaux algorithmes.

Dans ce chapitre, nous avons fait un état de l'art des algorithmes existants pour générer des suites à faible dispersion de taille quelconque. Les premiers algorithmes génèrent des points de manière séquentielle, *i.e.* les uns après les autres. Nous avons montré expérimentalement que cette méthode n'est pas globalement optimale : il est préférable de générer des points en mode « batch », *i.e.* l'ensemble des points à la fois. L'utilisation du critère du *minimax*, alias de la dispersion, n'est pas aisée algorithmiquement et peu d'algorithmes l'utilisent. L'utilisation du critère du *maximin* permet de simplifier les algorithmes. Parmi les algorithmes qui l'utilisent, nous pouvons distinguer les algorithmes basés sur des processus déterministes et ceux basés sur des processus stochastiques. Cependant, même si ces procédures de génération délivrent de bons résultats dans la majorité des cas, lorsque la taille de la suite n'est pas adéquate, les résultats délivrés ne sont pas des grilles de Shukarev. Nous avons alors proposé un nouvel algorithme de recuit-simulé qui, en pratique, délivre de bons résultats.

Le problème de générer $n$ points à faible dispersion a été traité dans ce chapitre. Le problème d'ajouter $n'$ points à une suite existante de $n$ points en minimisant la dispersion est un problème qui peut se rencontrer en pratique : il sera plus optimal d'ajouter ces $n'$ points simultanément que de manière itérative. L'algorithme que nous avons proposé est capable d'ajouter des points à une configuration initiale en minimisant la dispersion. Délivrant de bonnes solutions en pratique, il n'a cependant pas été créé dans cette optique. Il serait alors intéressant de s'intéresser plus profondément à cette question et de proposer de nouveaux algorithmes.

Enfin, nous avons illustré expérimentalement, que la différence de dispersion entre suite aléatoires et grilles de Shukarev reste significative lorsque la dimension augmente, alors que la différence de discrépance s'amenuise très rapidement. Cela représente alors une autre manière importante de distinguer ces deux critères d'uniformité !

# DEUXIÈME PARTIE

## EXPLORATION ET EXPLOITATION DANS L'APPRENTISSAGE ACTIF DE VARIÉTÉS

# - CHAPITRE 6 -

# SÉLECTION DE POINTS EN APPRENTISSAGE ACTIF

## Sommaire



CE CHAPITRE réalise un état de l'art de la littérature sur les méthodes d'échantillonnage en apprentissage actif par sélection de points adaptées aux problèmes de classification binaire. Dans cette définition de l'apprentissage actif, communément appelé « pool-based active learning » ou « selective sampling », l'algorithme d'apprentissage ne peut observer qu'une partie restreinte $\mathscr{E}$ de l'espace des observations $\mathscr{X}$. Celui-ci se décompose en l'ensemble $\mathscr{L}$ (comme « Labelled ») constitué des points d'apprentissage étiquetés $(x,y) \in \mathscr{X} \times \mathscr{Y}$ et l'ensemble $\mathscr{U}$ (comme « Unlabelled ») des points $x \in \mathscr{X}$ non étiquetés. L'ensemble $\mathscr{U}$ est un ensemble de cardinalité finie dans le cadre de la sélection de points ou d'instances, et il est possible de faire appel à un oracle expert pour étiqueter toute instance de $\mathscr{U}$. On a alors les relations suivantes : $\mathscr{E} = \mathscr{L} \cup \mathscr{U}$ et $\mathscr{U} \cap \mathscr{L} = \oslash$.

Nous notons $\mathscr{M}$ le un modèle prédictif que l'on cherche à apprendre dans un espace $\mathbb{M}$. Dans ce chapitre, le mot modèle est un synonyme d'hypothèse. En effet, à un modèle $\mathscr{M} \in \mathbb{M}$ correspond une hypothèse $h \in \mathbb{H}$ et inversement : il existe une bijection entre les modèles et les hypothèses. **Nous faisons cet amalgame car, dans la communauté scientifique de l'apprentissage actif, il est d'usage d'utiliser le terme modèle plutôt qu'hypothèse.**
Ce chapitre se situe bien dans la partie sur l'exploration et l'exploitation du modèle. En effet, nous postulons qu'il existe déjà un premier modèle $\mathscr{M}$ et nous cherchons à l'améliorer.

Muslea *et al.* (2002) ont défini de manière formelle le problème de l'échantillonnage sélectif. Dans leur définition, l'algorithme d'apprentissage parcourt l'ensemble $\mathscr{U}$ des instances candidates à l'étiquetage et, pour chaque instance $u \in \mathscr{U}$, utilise une fonction d'utilité $Utile(u, \mathscr{M})$ qui estime l'intérêt de cette instance pour le modèle $\mathscr{M}$. L'algorithme présente alors à l'oracle expert l'instance la plus utile pour le modèle $\mathscr{M}$. Une fois étiquetée, celle-ci est rajoutée à l'ensemble $\mathscr{L}$ des points d'apprentissage de l'algorithme. En effectuant de manière itérative cette procédure, l'algorithme doit pouvoir améliorer l'hypothèse apprise par le modèle.

Trois critères d'arrêts peuvent être utilisés :
- l'ensemble $\mathscr{U}$ des instances non étiquetées est vide ;
- les instances non étiquetées ont toutes une utilité nulle ou faible ;
- un nombre maximal de présentations d'instances à l'oracle est atteint.

Cette stratégie est écrite en pseudo-code dans l'algorithme 6.0.1.

**Algorithme 6.0.1** : Algorithme d'échantillonnage sélectif formalisé par Muslea *et al.* (2002).

**Entrées** :
- Modèle $\mathscr{M} \in \mathbb{M}$ prédictif ;
- Ensemble $\mathscr{L}$ d'instances étiquetées ;
- Ensemble $\mathscr{U}$ d'instances non étiquetées ;
- La fonction $Utile : \mathscr{X} \times \mathbb{M} \to \mathbb{R}$ , qui, à toute instance non étiquetée $x \in \mathscr{X}$ et tout modèle $\mathscr{M} \in \mathbb{M}$ associe l'utilité de l'instance pour l'apprentissage.

**répéter**

1. Apprendre le modèle grâce à l'algorithme $\mathscr{A}$ et l'ensemble $\mathscr{L}$ des exemples étiquetés ;
2. Rechercher l'instance $q^* = \arg\max_{x \in \mathscr{U}} Utile(x, \mathscr{M})$ ;
3. Ôter $q^*$ de $\mathscr{U}$, le présenter à l'expert pour obtenir son étiquette $f(q^*)$, et rajouter le couple $(q^*, f(q^*))$ à l'ensemble $\mathscr{L}$.

**jusqu'à** *Critère d'arrêt* ;

**Sorties** :
- Modèle $\mathscr{M} \in \mathbb{M}$ prédictif ;
- Ensemble $\mathscr{L}$ d'instances étiquetées ;
- Ensemble $\mathscr{U}$ d'instances non étiquetées.

Cet algorithme est générique. En effet il est seulement nécessaire de définir la fonction $Utile$ d'utilité d'un point pour définir une nouvelle méthode d'apprentissage actif par sélection de points. Cet algorithme permet ainsi d'identifier et d'uniformiser la classe des algorithmes de sélection de points. Cependant, avant cette identification, des algorithmes basés sur ce principe ont été développés. Dans la suite de ce chapitre, nous allons présenter ces différents algorithmes de sélection de points, chacun utilisant une fonction d'utilité différente et nous identifierons cette dernière. Nous commencerons par présenter l'échantillonnage par incertitude, puis l'échantillonnage par réduction de l'erreur en généralisation, nous continuerons par l'échantillonnage par comité de modèles et l'échantillonnage par réduction de l'espace des versions par arbre de décision. Enfin, nous présenterons l'échantillonnage actif bayésien.

## 6.1 Sélection de points par incertitude

La sélection de points par incertitude, ou *pool-based active learning*, a été introduite par Thrun et Moeller (1993) et reprise ensuite par Lewis et Gale (1994). Celle-ci a pour fonction d'utilité l'incertitude des prédictions du modèle.

La sélection des exemples se fait suivant l'algorithme :
- le modèle prédit les étiquettes de l'ensemble des points candidats ;
- le modèle estime l'incertitude de chaque prédiction ;
- le modèle sélectionne les points les plus incertains et les présente à l'oracle.

La caractéristique importante de cet échantillonnage est donc la possibilité du modèle d'estimer l'incertitude de ses prédictions. Nous pouvons alors distinguer deux types d'estimation : l'estimation basée sur la probabilité de classe prédite, et l'estimation par rapport à la distance de la frontière de classification apprise.

**Mesure d'incertitude basée sur la probabilité de la classe prédite :** lorsque le modèle prédit la classe d'une nouvelle instance et qu'il est capable de prédire également la probabilité de cette classe, nous pouvons alors définir une incertitude basée sur celle-ci. Dans ce cas, le modèle prédit la classe $y$ la plus probable avec la probabilité $\mathbb{P}_{\mathscr{M}}[y|x]$. Lorsque cette probabilité est élevée, le modèle a une grande confiance en sa prédiction, et l'instance n'a pas à être sélectionnée pour présentation à l'oracle. A l'inverse, lorsque cette probabilité est faible, le modèle a une faible confiance en sa prédiction et il est préférable que l'instance soit présentée à l'expert pour étiquetage. L'incertitude d'une instance est donc :

$$Incertitude(x) = \frac{1}{\arg\max_{y \in \mathscr{Y}} \mathbb{P}_{\mathscr{M}}[y|x]}; x \in \mathscr{U}.$$

Cette mesure d'incertitude se base sur la capacité de l'algorithme à estimer la probabilité des classes. Dans la pratique, peu de classes d'algorithmes sont capables de les fournir. Les SVMs, « Support Vector Machine », ou « Séparateurs à Vaste Marge » (voir Shawe-Taylor et Cristianini (2000)) sont capables de délivrer une estimation assez grossière de cette probabilité. En se basant sur les réseaux bayésiens, cet algorithme a été modifié par Tipping (2001) pour définir les « Relevance Vector Machine » qui sont capables de délivrer une meilleure estimation des probabilités. Lindenbaum *et al.* (2004) proposent un algorithme utilisant cette méthode avec la règle des $k$ plus proches voisins. Enfin, la plus grande famille d'algorithmes délivrant ces probabilités est la famille des algorithmes bayésiens.

**Mesure d'incertitude basée sur la distance à la frontière :** l'incertitude de la prédiction d'un point est définie par la distance à la frontière de classification fournie par le modèle $\mathscr{M}$. Plus l'instance sera prédite proche de la frontière de décision, plus celle-ci sera incertaine. En notant $\text{Front}_{\mathscr{M}}$ la frontière de classification fournie par le modèle, l'incertitude d'une instance $x$ est donc :

$$Incertitude(x) = \frac{1}{d(x, \text{Front}_{\mathscr{M}})}; x \in \mathscr{U}$$

Ceci suppose que l'on soit facilement capable d'estimer cette distance entre cette instance et cette frontière apprise par le modèle. Dans la majorité des algorithmes d'apprentissage, on peut estimer cette distance par approximation numérique. Cependant, dans la configuration où nous disposons de beaucoup d'instances candidates proches les unes des autres dans l'espace $\mathscr{X}$, cela requerra un nombre de calculs importants afin d'estimer efficacement la distance et donc l'incertitude. Cela limite donc l'application de cette méthode dans ces cas.

**Limites de la sélection de points par incertitude :** cette stratégie est facile à mettre en œuvre car, à chaque itération, le nombre d'exemples à étiqueter est égal au nombre d'instances présentes dans l'ensemble $\mathscr{U}$. En pratique, pour pouvoir utiliser cette méthode, il est nécessaire, soit d'utiliser une classe d'algorithmes capables de fournir les probabilités, soit d'être capable d'estimer la distance entre une instance et la frontière délivrée par le modèle. La première condition restreint le nombre d'algorithmes utilisables. La deuxième condition implique souvent un nombre de calculs important et ralentit donc le processus.
Lorsque les données ne sont pas séparables par le modèle, cette procédure va surtout sélectionner les points qui sont présents dans les zones de mélange des classes. Cela est le cas, par exemple, lorsque l'espace d'hypothèses $\mathbb{M}$ n'est pas suffisamment riche. De la même manière, l'algorithme va sélectionner principalement les instances proches des frontières de décisions connues. Des zones entières de l'espace $\mathscr{X}$ peuvent alors être littéralement occultées. Par conséquent, cette stratégie n'exploite que localement le modèle et favorise ainsi l'exploitation du modèle au détriment de son exploration. Une alternative de cet algorithme a été proposée par Nguyen et Smeulders (2004) en utilisant une méthode de clustering.

**Sélection de points par incertitude avec clustering :** pour favoriser l'exploration du modèle cible, et pas uniquement son exploitation, Nguyen et Smeulders (2004) réalisent un clustering sur l'ensemble $\mathscr{L} \cup \mathscr{U}$ des données étiquetées et non étiquetées à la manière des cartes « SOM » (Self Organizing Map). En faisant l'hypothèse que les centroïdes sont bien représentatifs des clusters et que les éléments d'un cluster ont les mêmes étiquettes, ils présentent les centroïdes de ces clusters à l'oracle. D'une manière équivalente à l'algorithme de Roy et McCallum (2001) (voir paragraphe 6.2), il est possible d'estimer la contribution de chaque centroïde à l'erreur empirique d'apprentissage. Nous sélectionnons le centroïde ayant la plus forte contribution : celui-ci possède alors, par propriété, le plus d'exemples diversifiés. Puis nous présentons les instances de ce centroïde à l'oracle expert et les intégrons à l'ensemble $\mathscr{L}$ d'apprentissage du modèle.
Cette étape de clustering est réalisée à chaque itération de l'échantillonnage. En utilisant, des clusters de taille importante, *i.e.* lorsqu'il y a peu de clusters, les centroïdes possèdent une grande dispersion et sont donc éloignés, les clusters possèdent beaucoup de données, et on favorise ainsi l'exploration du modèle. A l'inverse, en utilisant des clusters de petite taille, *i.e.* qu'il existe beaucoup de clusters, on favorise alors l'exploitation du modèle. En commençant par des clusters de taille élevée, et en faisant diminuer celle-ci au cours des itérations, on peut assurer un équilibre entre exploration et exploitation des données.
Enfin, remarquons que cette méthode étiquette plusieurs instances en même temps (celles du centroïde sélectionné), ce qui peut être restrictif lorsqu'il existe un nombre maximal de sollicitations de l'oracle, celui-ci pouvant être rapidement atteint.
De plus, une hypothèse fondamentale et non négligeable de l'algorithme de Nguyen et Smeulders (2004) réside dans le fait que l'on puisse étiqueter le centroïde choisi. Cela suppose que l'oracle expert soit capable d'étiqueter tout point de $\mathscr{X}$ qu'on lui présente. Cette méthode est donc à la frontière entre échantillonnage sélectif et échantillonnage génératif. Dans le cas où seules les instances de l'ensemble $\mathscr{U}$ peuvent être étiquetées, cette méthode n'est pas utilisable.
Auparavant, Lyhyaoui *et al.* (1999) ont déjà utilisé cette méthode pour sélectionner les instances les plus informatives avec un modèle utilisant des RBF (Radiales Basis Functions) et l'algorithme des SVMs.

**Sélection de points par incertitude avec comité de modèles :** les méthodes présentées précédemment sélectionnent l'instance à étiqueter en se basant sur l'utilité de celle-ci via son incertitude. Afin d'obtenir une « meilleure » estimation de l'utilité, Lu *et al.* (2010) proposent d'utiliser plusieurs modèles capables de fournir cette estimation, et de sélectionner celle qui maximise le désaccord entre ces estimations.

**Sélection de points par incertitude de Fisher :** Schein et Ungar (2007) présentent une méthode similaire basée sur un critère d'information de l'ensemble d'apprentissage. En utilisant des méthodes de régression logistique pour la classification, ils proposent de sélectionner les instances qui réduisent au mieux un critère d'« A-optimalité » qui est la trace de l'inverse de la matrice d'informations de Fisher (voir Fisher, 1951) de l'ensemble d'apprentissage.

## 6.2 Sélection de points par réduction de l'erreur en généralisation

Cohn *et al.* (1996) proposent de sélectionner les instances qui minimiseraient la variance des prédictions de tous les points de l'espace $\mathscr{X}$. Cette variance représente une fonction de coût particulière. Nous pouvons reprendre leur approche et la généraliser avec d'autres fonctions de coût : nous sélectionnons les instances qui minimisent l'erreur en généralisation du modèle.

Nous rappelons qu'il existe, dans ce chapitre, un amalgame entre modèle et hypothèse. En effet, à un modèle $\mathscr{M} \in \mathbb{M}$ correspond une hypothèse $h \in \mathbb{H}$ et inversement. Il existe donc une bijection entre les modèles et les hypothèses, et il est alors possible de redéfinir pour un modèle les erreurs en généralisation et empirique définies précédemment pour une hypothèse au chapitre 1.1 (page 8) par les équations 1.1 et 1.2. En utilisant la définition des fonctions de coût $l$ adaptée aux modèles, l'erreur en généralisation est définie à une itération $t$ par :

$$\mathscr{R}_t\left[\mathscr{M}\right] = \int_{x \in \mathscr{X}} l\left(\mathscr{M}, x\right) \mathscr{P}\left(x\right) dx$$

Le principe de l'algorithme est le suivant : pour chaque instance candidate $x^{new} \in \mathscr{X}$, et pour chaque étiquette possible $y^{new} \in \mathscr{Y}$, l'algorithme prédit le nouveau modèle $\mathscr{M}_{(x^{new}, y^{new})}$ appris sur l'échantillon d'apprentissage $\mathscr{L} \cup (x^{new}, y^{new})$. Cependant, l'étiquette $y^{new}$ associée à l'instance $x^{new}$ n'est pas connue. L'erreur en généralisation à l'itération $t+1$ se fait alors en intégrant ces erreurs suivant toutes les classes possibles de $\mathscr{Y}$ pondérées de leur probabilité d'apparition. L'erreur en généralisation à l'itération $t+1$ devient donc :

$$\mathscr{R}_{t+1}\left[\mathscr{M}_{x^{new}}\right] = \int_{y \in \mathscr{Y}} \mathscr{P}\left(y | x^{new}\right) dy \int_{x \in \mathscr{X}} l\left(\mathscr{M}_{(x^{new}, y)}, x\right) \mathscr{P}\left(x\right) dx$$

L'algorithme sélectionne alors l'instance $q \in \mathscr{U}$ qui possède l'erreur en généralisation la plus faible, *i.e.* $q = \underset{x^{new} \in \mathscr{U}}{\arg\min} \mathscr{R}_{t+1}\left[\mathscr{M}_{x^{new}}\right]$. Une fois cette instance $q^*$ sélectionnée, l'algorithme la retire de l'ensemble $\mathscr{U}$, la présente à l'expert pour obtenir son étiquette $y_{q^*}$ et ajoute le point $(q^*, y_{q^*})$ à l'ensemble $\mathscr{L}$.

**Utilisation en pratique de cet algorithme :** en pratique, il n'est pas possible d'utiliser cet algorithme. En effet, nous ne connaissons pas la distribution *a priori* de $y$ sachant $x$. Une manière de contourner ce problème consiste à mettre sur les classes de $\mathscr{Y}$ une distribution uniforme. Une autre méthode consiste à estimer cette probabilité sur l'ensemble des points de $\mathscr{L}$ étiquetés.
En supposant ce problème résolu, nous sommes alors confrontés au calcul de la fonction de coût sur l'espace $\mathscr{X}$ ( voir équation (6.2)). Par conséquent, le calcul de cette erreur en généralisation n'est pas possible. Une alternative consiste donc à estimer cette erreur en généralisation par l'erreur empirique sur les données d'apprentissage. Cette stratégie est celle proposée par Roy et McCallum (2001) dans le cadre de la classification de textes. Avec un *a priori* uniforme sur la distribution de $\mathscr{P}\left(x\right)$, l'erreur de généralisation est estimée avec les points de $\mathscr{L}$ disponibles à l'itération $t$, par l'erreur empirique :

$$\hat{\mathscr{R}}_t\left(\mathscr{M}\right) = \frac{1}{\#\left(\mathscr{L}\right)} \sum_{i=1}^{\#\left(\mathscr{L}\right)} l\left(\mathscr{M}, x_i\right) \text{ où } x_i \in \mathscr{L}$$

Pour chaque instance $x^{new}$ de $\mathscr{U}$ candidate à l'étiquetage, l'algorithme d'apprentissage apprend le modèle pour les différentes valeurs possibles des étiquettes $y \in \mathscr{Y}$. Une fois tous ces apprentissages réalisés, nous pouvons estimer l'erreur empirique $\hat{\mathscr{R}}_{t+1}(\mathscr{M}_{x^{new}})$ en utilisant une estimation $\hat{\mathscr{P}}(y|x)$ de $\mathscr{P}(y|x)$. L'algorithme sélectionne alors l'instance ayant l'erreur empirique minimale et la présente à l'expert pour étiquetage.

Les conditions d'arrêt de l'algorithme sont les mêmes que celles de l'algorithme précédent, à savoir :

- l'ensemble $\mathscr{U}$ des instances non étiquetées est vide ;
- les instances non étiquetées ont toutes une utilité nulle ou faible ;
- un nombre maximal de présentations d'instances à l'oracle est atteint.

Une écriture en pseudo-code de cet algorithme est présentée à l'algorithme (6.2.1).

**Limites de la sélection d'instances par réduction de l'erreur en généralisation :** cette stratégie explore toutes les instances avec toutes les étiquettes possibles, ce qui en fait une stratégie très exhaustive. Cependant, elle est très gourmande en calculs, car à chaque itération, la sélection des instances à étiqueter entraîne $\#(\mathscr{U}) \times \#(\mathscr{Y})$ apprentissages du modèle, auxquels il faut ajouter ensuite l'estimation sur les données d'apprentissage des modèles obtenus.
Cette stratégie possède des versions différentes, notamment par la diversité des fonctions de coût utilisées. Une autre différence concerne l'estimation de la qualité des modèles : à la manière de la procédure de validation croisée, nous pouvons apprendre sur une partie des données de l'ensemble d'apprentissage et tester les modèles sur une autre partie de ces données. En répliquant cette procédure pour l'estimation des erreurs d'apprentissage et en les moyennant, nous pouvons alors limiter le risque de sur-apprentissage du modèle.

Dans cette approche, pour reprendre la notion d'utilité introduite par Muslea *et al.* (2002) dans l'algorithme 6.0.1, l'utilité d'une instance est égale à l'inverse de l'erreur en généralisation $\mathscr{R}_{t+1}[\mathscr{M}_{x^{new}}]$ du modèle. Celle-ci est estimée par l'erreur empirique $\hat{\mathscr{R}}_{t+1}[\mathscr{M}_{x^{new}}]$ sur les données d'apprentissage.

## 6.3 Sélection de points par réduction de l'espace des versions

En apprentissage, l'espace d'hypothèses $\mathbb{H}$ est l'espace fonctionnel dans lequel le modèle initial d'apprentissage $\mathscr{M}$ sélectionne le modèle (ou par bijection l'hypothèse $\hat{f}$) qui correspond au mieux à la fonction cible par rapport aux données d'apprentissage. Parmi toutes les hypothèses de $\mathbb{H}$, certaines prédisent convenablement toutes les données d'apprentissage, *i.e.* l'ensemble $\mathscr{L}$ : ces hypothèses sont appelées consistantes. L'ensemble de toutes les hypothèses consistantes est donc un sous-ensemble de $\mathbb{H}$ et s'appelle l'espace des versions. Initialement, l'espace des versions est l'espace $\mathbb{H}$ dans son intégralité. Avec les premières données d'apprentissage, le modèle $\mathscr{M}$ ne va sélectionner que les versions consistantes.

Lorsque l'on présente un nouvel exemple étiqueté aux fonctions appartenant à cet espace de versions, certaines hypothèses ne vont plus rester consistantes, ce qui, par conséquent, réduit immédiatement cet espace. Le modèle choisi (ou l'hypothèse choisie) prendra la place de l'ancien et deviendra $\mathscr{M}$. La méthode de réduction de l'espace des versions consiste donc à sélectionner les instances qui permettront de réduire la richesse de l'espace des versions.

Nous présentons dans la suite deux méthodes de sélection de points par réduction de l'espace des versions : celle par comités de modèles et celle sans comité.

**Algorithme 6.2.1** : Algorithme de sélection d'instances par réduction de l'erreur empirique de généralisation proposé par Roy et McCallum (2001).

**Entrées** :
- Modèle $\mathscr{M} \in \mathbb{M}$ ;
- Ensemble $\mathscr{L}$ d'instances étiquetées ;
- Ensemble $\mathscr{U}$ d'instances non étiquetées ;
- $\mathscr{Y}$ l'ensemble des étiquettes possibles ;
- Une fonction d'évaluation de $\hat{\mathscr{R}}_{t+1} : \mathscr{U} \times \mathbb{M} \rightarrow \mathbb{R}$ qui estime l'erreur empirique du modèle $\mathscr{M}$, à l'itération $t$ appris sur les données d'apprentissage $\mathscr{L} \cup (f, f(x))$.

**répéter**

1. Apprendre le modèle grâce à l'ensemble $\mathscr{L}$ des exemples étiquetés ;
2. **pour** *chaque instance* $x^{new} \in \mathscr{U}$ **faire**
   - **pour** *chaque label* $y \in \mathscr{Y}$ **faire**
     - Apprendre le modèle $\mathscr{M}$ sur l'ensemble des données d'apprentissage $\mathscr{L} \cup (x^{new}, y)$
     - Calculer l'erreur de généralisation attendue $\hat{\mathscr{R}}_{t+1}\left(\mathscr{M}_{(x^{new},y)}\right)$

     **fin**
   - Calculer l'erreur de généralisation espérée $\hat{\mathscr{R}}_{t+1}(\mathscr{M}_{x^{new}}) = \sum_{y \in \mathscr{Y}} \hat{\mathscr{R}}_{t+1}\left(\mathscr{M}_{(x^{new},y)}\right) \hat{\mathscr{P}}(y|x^{new})$

   **fin**
3. Rechercher l'instance $q^* = \arg\max_{x^{new} \in \mathscr{U}} \hat{\mathscr{R}}_{t+1}(\mathscr{M}_{x^{new}})$ ;
4. Ôter $q^*$ de $\mathscr{U}$, le présenter à l'expert pour obtenir son étiquette $f(q^*)$, et rajouter le couple $(q^*, f(q^*))$ à l'ensemble $\mathscr{L}$.

**jusqu'à** *Critère d'arrêt* ;

**Sorties** :
- Modèle $\mathscr{M} \in \mathbb{M}$ ;
- Ensemble $\mathscr{L}$ d'instances étiquetées ;
- Ensemble $\mathscr{U}$ d'instances non étiquetées.

### 6.3.1 Utilisation d'un comité de modèles

Seung *et al.* (1992) proposent d'utiliser plusieurs modèles d'apprentissage ayant chacun leur propre espace d'hypothèses pour sélectionner l'instance à étiqueter. Cette méthode est connue sous le nom de « Querry by Committee ». Par exemple, un premier modèle peut être un modèle de SVM (voir Shawe-Taylor et Cristianini (2000)) ayant pour espace d'hypothèses des combinaisons linéaires de fonctions noyaux, un deuxième modèle peut être un modèle de réseau de neurones avec pour espace d'hypothèses une combinaison de perceptrons linéaires, ou de fonctions radiales, etc.... A titre d'illustration, Abe et Mamitsuka (1998) proposent d'utiliser comme modèles des méthodes de « bagging » (« Querry by bagging ») et de « boosting » (« Querry by boosting »).
Tous ces modèles sont ensuite entraînés simultanément sur les mêmes données d'apprentissage de $\mathscr{L}$. En supposant que chaque modèle soit capable de délivrer une hypothèse $h$ consistante, le comité de modèles représente donc un échantillon d'hypothèses qui doit être représentatif de l'espace des versions. Nous présentons les différentes instances de $\mathscr{U}$ à ce comité de modèles, et le désaccord au sein du comité est mesuré. Les exemples qui possèdent le plus grand désaccord sont ceux qui ont la plus forte probabilité de réduire l'espace des versions s'ils étaient étiquetés.

**Mesure de désaccord par vote majoritaire :** Abe et Mamitsuka (1998) proposent de mesurer le désaccord pour une instance $x \in \mathscr{U}$ par l'effectif de modèles qui ont un vote différent du vote majoritaire du comité. En notant $\hat{f}(x)$ la prédiction pour le point $x \in \mathscr{U}$ du comité de modèles, et $\hat{f}_{\mathscr{M}_k}(x)$ la prédiction pour le point $x$ du modèle $\mathscr{M}_k$ et en ayant $m$ modèles dans le comité, le désaccord s'écrit de la manière suivante :

$$\text{Désaccord}(x) = \sum_{k=1}^{m} \mathbb{1}_{\left\{\hat{f}_{\mathscr{M}_k}(x) \neq \hat{f}(x)\right\}}$$

**Mesure de désaccord basée sur l'entropie :** en s'inspirant de la théorie de l'information, Freund *et al.* (1997) estiment le désaccord du comité de modèles en utilisant l'entropie des prédictions. Pour chaque instance candidate de $\mathscr{U}$, et pour chaque étiquette $y_i \in \mathscr{Y}$, nous pouvons estimer les probabilités conditionnelles $\mathscr{P}(y_i|x)$ de manière empirique avec les $m$ modèles du comité par : $\hat{\mathscr{P}}(y_i|x) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\{\hat{f}_{\mathscr{M}_i}(x)=y_i\}}$.
L'entropie des prédictions des modèles $\mathscr{M}_1,\ldots,\mathscr{M}_m$ est alors estimée pour le point $x \in \mathscr{U}$ par :

$$\mathscr{H}(x) = \sum_{i=1}^{\#(\mathscr{Y})} -\hat{\mathscr{P}}(y_i|x)\log\hat{\mathscr{P}}(y_i|x)$$

Plus l'entropie des prédictions du comité de modèles est grande, plus le désaccord entre les modèles prédictifs est important.

**Mesure de désaccord basée sur la divergence :** McCallum et Nigam (1998) proposent une autre mesure qui prend en compte la confiance que les modèles du comité ont pour mesurer leur désaccord. Cela suppose que les modèles du comité soient capables d'estimer ces probabilités. Pour un point $x$ de $\mathscr{U}$, pour un modèle $\mathscr{M}_k$ du comité, on peut mesurer la « Kullback Leiber divergence » entre la sortie probabiliste du modèle $k$ et la sortie probabiliste du comité de modèles $\mathscr{M}_1,\ldots,\mathscr{M}_m$ par :

$$\text{Div}\left(\mathscr{P}(y|x,\mathscr{M}_k);\mathscr{P}(y|x,\mathscr{M}_1,\ldots,\mathscr{M}_m)\right) = \sum_{i=1}^{\#(\mathscr{Y})} \mathscr{P}(y_i|x,\mathscr{M}_k)\log\left(\frac{\mathscr{P}(y_i|x,\mathscr{M}_k)}{\mathscr{P}(y_i|x,\mathscr{M}_1,\ldots,\mathscr{M}_m)}\right)$$

La mesure de désaccord utilisée est alors la moyenne des divergences sur les $m$ modèles du comité, soit :

$$\text{Désaccord}(x) = \sum_{k=1}^{m} \text{Div}\left(\mathscr{P}(y|x,\mathscr{M}_k);\mathscr{P}(y|x,\mathscr{M}_1,\ldots,\mathscr{M}_m)\right)$$

**Limite de la sélection par comité de modèles :** avant de commencer l'apprentissage, il est nécessaire de mettre en place le comité de modèles (choix des familles types d'apprentissage, *e.g.* réseau de neurones, SVMs, boosting,...) et de choisir la mesure de désaccord. A chaque itération de sélection de points, nous réalisons $m$ apprentissages et $\#(\mathscr{U}) \times m$ prédictions. Ce nombre d'opérations est peu important. Cependant, nous n'avons aucune garantie que cette méthode échantillonne correctement l'espace des versions. De plus, nous avons supposé que les modèles sélectionnent des hypothèses consistantes, ce qui est une hypothèse forte, pas nécessairement facile à garantir.

Dans cette approche, pour reprendre la notion d'utilité introduite par Muslea *et al.* (2002) dans l'algorithme (6.0.1), l'utilité d'une instance est égale à l'inverse de la mesure de désaccord du comité de modèles.

### 6.3.2 Sans utilisation d'un comité de modèles

Dans le cadre de la sélection d'instances, Tong et Koller (2000) et Schohn et Cohn (2000) proposent de sélectionner les instances qui, dans un apprentissage par SVMs, se situent dans la marge. Cette méthode permet de manipuler ainsi directement l'espace des versions sans avoir à utiliser un comité de modèles. On peut dire autrement que le comité de modèles est constitué uniquement de ce modèle. Campbell *et al.* (2000) généralisent cette méthode pour tous les classifieurs à marge dont les SVMs font partie. Ainsi, une instance faisant partie des marges du classifieur est assez proche de la frontière de classification apprise et de la frontière réelle. Plus l'instance est proche de la frontière apprise, plus la classification de l'instance par l'approximation est incertaine, et plus l'utilité de l'instance pour améliorer l'apprentissage est ainsi élevée. Auparavant, Cohn *et al.* (1994) utilisaient la même approche dans le cas des réseaux SG-nets.

Dasgupta *et al.* (2005) proposent d'introduire une distribution de probabilités sur l'espace des hypothèses, et de sélectionner l'instance qui permet d'obtenir, en fonction de l'étiquette attribuée, l'espace des versions le plus équiprobable au sens de la distribution introduite. En utilisant ces travaux, Bondu et Lemaire (2007b) considèrent que cette approche de sélection d'instances peut s'écrire comme le parcours d'un arbre de décision. Cette approche reste néanmoins assez théorique.

Dasgupta (2006) proposent d'utiliser un critère appelé *splitting index*. Cet index permet de sélectionner les instances qui, une fois étiquetées, permettraient de réduire exponentiellement la complexité de l'espace des versions. Hanneke (2007) étend cet index au cas de l'apprentissage en présence de bruit. Comme pour la dimension de Vapnik-Chervonenkis et les nombres de couvertures, le but est d'estimer la capacité d'apprentissage de l'espace d'hypothèses en rajoutant un nouvel exemple. Ces index ont alors pour objectif de sélectionner l'exemple réduisant au mieux la complexité de l'espace d'hypothèses.

## 6.4 Sélection de points par apprentissage de modèles locaux

En se basant sur la curiosité adaptative des robots définie par Oudeyer et Kaplan (2004), Bondu (2008) propose de scinder le problème d'apprendre sur l'espace $\mathscr{X}$ en sous-problèmes. Il partitionne itérativement l'espace en sous-espaces et entraîne alors localement un modèle local sur chacun de ces sous-espaces. Chaque modèle local $m_i$ est alors spécialisé dans une zone de l'espace des variables $\mathscr{X}$ et ne peut être entraîné que sur les données $l_i$ de $\mathscr{L}$ comprises dans cette zone. Cette méthode d'apprentissage par modèles locaux est une méthode qui donne un cadre d'application nouveau aux méthodes de sélection de points présentées précédemment : en effet, chaque sous-modèle sélectionne alors un point d'apprentissage à présenter à l'oracle expert en utilisant une de ces méthodes précédentes.

Dans cette sélection par modèles locaux, la première étape consiste à sélectionner le modèle local, *i.e.* la zone de l'espace $\mathscr{X}$, pour laquelle l'apport d'une nouvelle information serait le plus bénéfique. Cette sélection de zone est réalisée indépendamment de l'ensemble des instances candidates à l'étiquetage. Une fois cette zone sélectionnée, les méthodes présentées précédemment peuvent alors s'appliquer au modèle local, *i.e.* que l'on peut appliquer la sélection de l'instance parmi celles qui se situent dans la zone de définition du modèle local. La procédure d'apprentissage par modèles locaux et de sélection de points est présentée à l'algorithme (6.4.1).

**Algorithme 6.4.1** : Algorithme de sélection de points par modèles locaux.

**Entrées** :
- Ensemble de $n$ modèles locaux $\mathscr{M} = \{m_1, m_2, \ldots, m_n\}$ ;
- Ensemble $\mathscr{L} = \{l_1, l_2, \ldots, l_n\}$ d'instances étiquetées ;
- Ensemble $\mathscr{U} = \{u_1, u_2, \ldots, u_n\}$ d'instances non étiquetées ;

n=1 ;
**répéter**
1. Choisir le modèle local $m_i$ à améliorer ;
2. Choisir une instance $x^{new} \in u_i$ à présenter à l'oracle ;
3. Présenter cette instance afin d'obtenir son label $y^{new}$ ;
4. Retirer $x^{new}$ de $u_i$ et ajouter $(x^{new}, y^{new})$ à $l_i$ ;
5. Approfondir le modèle local $m_i$ et l'estimer ;
6. **si** *Critère de séparation* **alors**
   - Partitionner $l_i$ en deux sous-ensembles $l_j$ et $l_k$ ;
   - Dupliquer $m_i$ en deux modèles locaux $m_j$ et $m_k$ ;
   - n=n+1

   **fin**

**jusqu'à** *Critère d'arrêt* ;

**Sorties** :
- Ensemble de $n$ modèles locaux $\mathscr{M} = \{m_1, m_2, \ldots, m_n\}$ ;
- Ensemble $\mathscr{L} = \{l_1, l_2, \ldots, l_n\}$ d'instances étiquetées ;
- Ensemble $\mathscr{U} = \{u_1, u_2, \ldots, u_n\}$ d'instances non étiquetées.

La mise en œuvre de cette méthode amène alors à se poser les questions suivantes :

1. Quel modèle doit-on sélectionner pour augmenter la qualité de l'apprentissage ?
   **Comment sélectionner les exemples à étiqueter pour le modèle choisi ?**
2. Quel critère utiliser pour déterminer si l'on doit scinder un modèle en sous-modèles ou non ?
   Si on doit scinder le modèle, en combien de modèles devons-nous le faire ?
3. Quels modèles locaux utiliser ?

**Réponse à la question 1 : sélection du modèle local**

A chaque itération de l'algorithme, nous pouvons sélectionner le modèle local qui s'améliore le mieux en supposant que c'est celui-ci qui possède la plus grande capacité d'amélioration. Oudeyer et Kaplan (2004) proposent de sélectionner les modèles en estimant leurs performances grâce aux exemples locaux étiquetés et à une mesure de performance. Les variations de cette mesure sur une fenêtre temporelle permettent alors d'évaluer les progrès des modèles. Dans le cadre de l'apprentissage, Bondu et Lemaire (2007a) proposent d'utiliser l'aire sous la courbe ROC des modèles comme mesure de performance.
La sélection des instances de $u_i$ à présenter à l'expert peut être réalisée soit de manière aléatoire, soit en utilisant les méthodes présentées précédemment.

**Réponse à la question 2 : critère pour scinder ou non un modèle local**

La réponse à cette question, proposée par Oudeyer et Kaplan (2004), est une réponse empirique : ils proposent de scinder en deux un modèle local lorsque celui-ci est construit grâce à un échantillon de données dont la taille est supérieure à un certain seuil. Les modèles ainsi sélectionnés sont ceux qui sont construits avec le plus d'exemples, et sont donc ceux qui, nous pouvons l'espérer, s'améliorent le mieux. Cette heuristique de partitionnement du modèle basée sur une nombre maximal d'exemples, et non sur un nombre minimal, permet ainsi de limiter le risque de sur-apprentissage.
Lorsqu'un modèle est identifié pour être scindé, ils proposent de le scinder en deux sous-modèles. Chaque modèle est ensuite entraîné sur une sous-zone de l'espace du modèle initial. La scission doit être réalisée de façon à ce que les exemples étiquetés se répartissent équitablement dans les espaces des deux sous-modèles. Pour chaque variable de $\mathscr{X}$, et pour chaque valeur prise par celle-ci sur les exemples étiquetés et les exemples non-étiquetés candidats, nous pouvons estimer la pertinence du couple en calculant la variance de part et d'autre de ces valeurs des prédictions du modèle sur les données étiquetées. Le couple variable-valeur qui minimise cette variance est alors retenu. - Remarquons que nous pouvons également très bien envisager d'utiliser des combinaisons linéaires de ces variables et des valeurs, un peu comme à la manière des arbres CART de décisions obliques proposée par Cantu Paz et Kamath (2003). - L'élaboration de ces zones permet alors d'obtenir des modèles locaux ayant de bonnes propriétés. Néanmoins, il faut veiller à ce que les modèles locaux ne deviennent pas trop locaux, *i.e.* qu'il ne faut pas avoir un trop grand nombre de modèles locaux, ce qui impliquerait de grandes zones d'apprentissage par cœur. Le critère d'arrêt de la procédure est donc important.

**Réponse à la question 3 : modèles locaux à utiliser**

Il n'existe pas, à notre connaissance, de travaux théoriques permettant de répondre à cette question. En pratique, les modèles locaux utilisés sont issus d'une même classe de modèles. Seuls les fonctions de base ou les paramètres sont variables. Par exemple, Oudeyer et Kaplan (2004) utilisent des procédures de $k$-plus proches voisins, alors que Bondu et Lemaire (2007a) utilisent des réseaux de neurones de même famille Néanmoins, à l'instar des méthodes de comité de modèles, il est tout à fait envisageable d'utiliser des modèles de classes différentes.

Dans cette approche, pour reprendre la notion d'utilité introduite par Muslea *et al.* (2002) dans l'algorithme 6.0.1, l'utilité d'une instance n'est alors pas directe. Conditionnellement au modèle local associé, l'utilité d'une instance est celle relative à la méthode de sélection utilisée.

## 6.5 Sélection de points par méthodes d'apprentissage semi-supervisé

Afin d'améliorer les performances d'apprentissage actif, plusieurs travaux incluent une phase d'apprentissage non-supervisé dans le processus d'apprentissage : ce mélange donne alors naissance à l'apprentissage dit semi-supervisé. Nous pouvons citer par exemple Chapelle (2005) qui

utilise l'échantillonnage par réduction de l'erreur en généralisation avec un apprentissage par fenêtre de Parzen. Il améliore expérimentalement la procédure en intégrant de l'apprentissage non supervisé pour choisir les points à présenter à l'oracle. Zhu *et al.* (2003) obtiennent le même résultat en classification multi-classes en utilisant des combinaisons de fonctions gaussiennes sur la base classique des chiffres postaux. L'algorithme de sélection de points avec clustering de Nguyen et Smeulders (2004) présenté à la section 6.1 fait partie évidente des méthodes d'apprentissage actif semi-supervisé. Li et Zhou (2011) réalisent également un clustering des instances candidates pour la sélection d'instances avec un apprentissage utilisant l'algorithme des SVMs.
Tous ces travaux sont assez expérimentaux et sont très dépendants des méthodes d'apprentissage utilisées. Dans un article au titre révélateur « Unlabeled data : Now it helps, now it doesn't », Singh *et al.* (2008) réalisent un état de l'art des résultats théoriques et expérimentaux, et comparent théoriquement l'apprentissage semi-supervisé à l'apprentissage statistique classique.

Bondu *et al.* (2010) proposent d'adapter l'apprentissage semi-supervisé au cas de la sélection de points par apprentissage par modèles locaux. Ils s'intéressent notamment à améliorer la procédure qui permet de décider quand scinder un modèle en deux sous-modèles et comment le scinder. Leurs travaux se basent sur la méthode de discrétisation supervisée avec un formalisme bayésien : la méthode MODL (Minimal Optimized Description Length) développée par Boullé (2006). Cette méthode permet de limiter le sur-apprentissage des modèles locaux. Leur stratégie cherche à sélectionner l'instance qui maximisera la qualité de futur modèle, sans connaître son étiquette et sans connaître le meilleur modèle à l'itération suivante parmi l'ensemble des modèles possibles. Cette approche est assez théorique et nécessite en pratique un temps de calculs important.

Zhou et Li (2005) combinent une méthode d'apprentissage non supervisé avec un comité de modèles. A la manière de l'apprentissage « co-training » défini par Blum et Mitchell (1998), ils entraînent trois classifieurs distincts pour choisir les instances à étiqueter et obtiennent un algorithme ayant de bonnes propriétés de généralisation. De plus, celui-ci ne possède pas beaucoup d'hypothèses et peut donc s'appliquer à un large panel de problèmes. Ils appliquent leur algorithme sur différents jeux de données de l'UCI et obtiennent des résultats encourageants. Cet travail est à la base d'un article (Zhou et Li (2010)) dans lequel ils font un état de l'art des méthodes semi-supervisées utilisant un comité de modèles. Ils les appliquent avec succès à la classification d'images et élargissent leur approche aux problèmes de régression. Cheng et Wang (2007) utilisent également deux modèles de SVMs pour former un algorithme dit « Co-SVM » pour sélectionner l'instance la plus utile au modèle, et appliquent cette méthode à de la classification d'images.

## 6.6 Analyse théorique des performances de ces méthodes

Supposons que l'on souhaite apprendre en dimension 1 la fonction de classification binaire suivante : $f(x,\theta) = \begin{cases} 1 & \text{si } x > \theta \\ 0 & \text{sinon} \end{cases}$. Dans le cadre de l'apprentissage statistique PAC classique, sous la condition que la fonction cible appartienne à l'espace d'hypothèses, pour un taux d'erreur $\varepsilon$ fixé, le nombre d'instances nécessaires à étiqueter est en $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$. Avec une approche de type sélection de points, en supposant que les instances candidates à l'étiquetage soient nombreuses et bien réparties sur la droite, le nombre d'instances sélectionnées pour étiquetage est en $\mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ pour la même précision $\varepsilon$.
La généralisation de ce résultat à des dimensions supérieures, à des fonctions cibles plus complexes et donc plus proches de problématiques réelles, fait actuellement l'attention de travaux théoriques en active learning.

Freund *et al.* (1997) ont analysé théoriquement l'algorithme de « querry-by-committee » présenté à la section (6.3.1). Sous une hypothèse bayésienne, ils montrent qu'il est possible d'obtenir un taux d'erreur en généralisation de $\varepsilon$ en disposant d'un nombre d'instances candidates de l'ordre de $\mathcal{O}\left(\frac{\mathcal{VC}}{\varepsilon}\right)$ et en n'en étiquetant qu'un nombre de l'ordre de $\mathcal{O}\left(\mathcal{VC}\log\left(\frac{1}{\varepsilon}\right)\right)$, où $\mathcal{VC}$ est la dimension de Vapnik-Chervonenkis de l'espace de la fonction cible. Ainsi, nous pouvons voir que la méthode de sélection d'instances permet, sous ces hypothèses, d'améliorer la procédure d'apprentissage. Ce résultat doit cependant être tempéré car les calculs réalisés par cet algorithme peuvent être complexes et nombreux dans certains cas. Gilad-Bachrach *et al.* (2006) utilisent cet algorithme couplé à des machines à noyaux et obtiennent de bonnes propriétés d'apprentissage en pratique.

Dasgupta *et al.* (2005) adaptent l'algorithme du perceptron et obtiennent un algorithme ayant la même complexité que le « Querry by Committee » présenté précédemment. Ils montrent ainsi un gain significatif sur le nombre d'instances à étiqueter par rapport au perceptron classique qui nécessite un nombre minimal d'appels à l'expert de l'ordre de $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$. Les deux différences principales entre leur approche et le « Querry by Committee » sont : (i) le « Querry by Committee » requiert une hypothèse bayésienne pour être étudié théoriquement, et (ii) « Querry by Committee » peut nécessiter un nombre de calculs important comparé au perceptron linéaire.

## 6.7 Conclusion et discussion

Les méthodes présentées dans ce chapitre permettent de sélectionner les instances candidates à l'étiquetage parmi un ensemble d'instances candidates. Elles sélectionnent les instances une par une et utilisent toutes une fonction d'utilité qui permet ainsi de les hiérarchiser entre elles. Nous avons choisi de présenter ces méthodes selon les fonctions d'utilité qu'elles utilisent, définies au début de ce chapitre. Une autre présentation possible aurait été d'utiliser le caractère collégial ou non de la décision : certaines méthodes n'utilisent qu'un seul modèle prédictif, d'autres méthodes font appel à un collège de modèles. Il est envisageable également de combiner plusieurs de ces méthodes pour prendre les décisions de présentation à l'expert.

Dans ce contexte d'apprentissage actif, le « pool-based learning », nous disposons entièrement de l'ensemble des instances candidates à chaque instant. Cela suppose que l'ensemble des instances est de cardinalité sensiblement supérieure à la cardinalité des exemples étiquetés. Aucune notion de dynamique de l'ensemble des instances n'est prise en compte. La stratégie de sélection des instances évalue toute la collection d'instances et donne une mesure d'utilité à chaque instance avant de sélectionner les meilleures à présenter à l'oracle. Cette théorie de sélection d'instances ne s'est pas construite d'une seule lignée, mais s'est développée petit à petit avec la résolution de problèmes réels d'apprentissage dans de nombreux domaines : on peut citer par exemple des problèmes de classification de textes (Lewis et Gale (1994), McCallum et Nigam (1998), Tong et Koller (2000), Hoi *et al.* (2006)), d'extraction d'informations (Settles et Craven (2008)), de classification d'images (Zhang et Chen (2002)), de classification de vidéos (Yan *et al.* (2003)), de reconnaissance de paroles (Tür *et al.* (2005)) ou bien de classification de gènes pour des diagnostics médicaux tels que le cancer par exemple (Liu (2004)). Néanmoins, cette méthode requiert de bonnes capacités de calculs et de stockages d'information, ce qui peut limiter en pratique son utilisation lorsque l'on dispose de peu de capacité de calculs. Ceci est souvent le cas lorsque les moyens de calculs sont mobiles ou embarqués sur des robots.

Lorsque les données sont disponibles en ligne, *i.e.* qu'elles arrivent en temps réel, et que l'on ne peut étiqueter qu'un nombre fini d'instances, ce problème de stockage et de choix des instances se pose aussi. Celui-ci se résout souvent de manière récursive. Cette méthode alternative appelée « stream based active learning » ou « sequential active learning », est détaillée en pratique dans la

thèse de Settles (2009).

La problématique initiale de cette thèse consiste à générer les points d'apprentissage, *i.e.* que l'on suppose que l'oracle est capable d'étiqueter tout point de l'espace, et en sollicitant l'oracle le moins de fois possible. Cette approche peut être vue comme un passage à la limite de l'ensemble des instances. Cependant, pour les raisons évoquées ci-dessus, cela n'est pas réalisable : les capacités de calcul ne sont pas infinies. Il convient donc de mettre en place une autre stratégie pour présenter à l'oracle les instances à étiqueter. Nous présentons une nouvelles stratégie adaptée au problème de classification binaire dans le chapitre suivant.

# - CHAPITRE 7 -

# EXPLORATION ET EXPLOITATION DANS UN ALGORITHME D'APPROXIMATION DE VARIÉTÉS

## Sommaire



DANS CE CHAPITRE, nous nous intéressons à l'apprentissage actif de variétés avec un échantillonnage adaptatif. L'échantillonnage adaptatif peut être vu comme un passage à la limite de l'échantillonnage sélectif présenté au chapitre précédent dans le cas où la base d'instances candidates à la présentation à l'oracle croît infiniment. En pratique, ce passage à la limite n'est pas réalisable car cela engendre un nombre de calculs infaisable et qui sont souvent redondants. L'échantillonnage adaptatif est une approche différente, qui vise à résoudre ce problème.

Dans la littérature d'apprentissage actif, l'échantillonnage adaptatif n'a pas été beaucoup étudié pour l'apprentissage de variétés. Ceci peut paraître surprenant car le nombre d'applications de celui-ci est relativement important.

Dans une première partie, nous présentons la principale étude dans la littérature sur l'apprentissage actif de variétés. Dans une deuxième partie, nous présentons un nouvel algorithme d'apprentissage actif pour exploiter une approximation de variétés sous une hypothèse de régularité. A partir d'une grille régulière, cet algorithme affine localement une grille régulière d'apprentissage. Dans une troisième et dernière partie, nous concluons.

Les travaux présentés dans ce chapitre sont théoriques et ont été réalisés en collaboration avec Olivier TEYTAUD.

## 7.1 Apprentissage actif de variétés

En apprentissage actif, les premières études théoriques des bornes d'erreur en approximation de variétés ont été réalisées dans deux cas particuliers. Le premier cas particulier est celui des problèmes qui étaient de dimension un, *e.g.* Burnashev et Zigangirov (1974) ou Hall et Molchanov (2003). Le deuxième type de cas particuliers concerne les problèmes multidimensionnels qui peuvent être résolus par une succession de problèmes unidimensionnels, *e.g.* Korostelev (1999) en analyse d'images.

Par la suite, Castro *et al.* (2005) étudient plus en détail les équations d'active learning en dimension quelconque. Leur étude est basée initialement sur l'influence d'un taux de bruit dans la base d'apprentissage, mais ces résultats sont valables dans le cas non bruité. Les équations présentées dans ce présent chapitre sont adaptées au contexte d'absence de bruit.
Afin de qualifier la régularité des frontières de classification, les auteurs utilisent les définitions suivantes :

1. une fonction $f : [0,1]^s \to \mathbb{R}$ est dite localement constante en un point $x$ si :
$$\exists \varepsilon > 0 : \forall y \in [0,1]^s : \|x-y\| < \varepsilon \Rightarrow f(y) = f(x);$$
2. une fonction $f : [0,1]^s \to \mathbb{R}$ est constante par morceaux si elle est bornée uniformément et localement constante en tout point $x$ de $[0,1]^s$ privé de $\partial f$ où $\partial f$ est un ensemble de $[0,1]^s$ de dimension $s-1$. De plus, pour tout rayon $r > 0$ fixé, le nombre minimal $N(r)$ de boules de rayon $\frac{r}{2}$ recouvrant la frontière $\partial f$ satisfait l'équation suivante :
$$\exists \beta > 0, N(r) \leq \beta \frac{1}{r^{s-1}}.$$

En utilisant toutes les hypothèses présentées précédemment, avec $N$ exemples d'apprentissage, les auteurs énoncent les résultats suivants avec le principe du *minimax* :

1. en apprentissage passif, ou apprentissage statistique, à des constantes multiplicatives près, la meilleure erreur en généralisation que l'on peut espérer est supérieure à $N^{-\frac{1}{s}}$ et inférieure à $\left(\frac{N}{\log N}\right)^{-\frac{1}{2}}$ ;
2. en apprentissage actif, à des constantes multiplicatives près, l'erreur en généralisation est supérieure à $N^{-\frac{1}{s-1}}$.

Enfin, dans leur contexte d'apprentissage actif, les auteurs montrent également que l'erreur en généralisation est bornée, à une constante multiplicative près, par : $\left(\frac{N}{\log N}\right)^{-\frac{1}{s-1+1/s}}$.

## 7.2 Un algorithme d'apprentissage actif pour approcher des variétés avec *a priori*

Dans ce chapitre, en supposant toujours qu'un oracle expert est capable de nous délivrer sans erreur l'étiquette de tout point $x \in I^s$, nous présentons une nouvelle procédure d'apprentissage actif dans le domaine $I^s = [0,1]^s$. Le but est toujours d'approximer une variété en faisant appel le moins de nombre de fois possibles à l'oracle. Cependant, nous n'utilisons pas la même définition de régularité que dans les travaux de Castro *et al.* (2005).

Nous nous plaçons dans le cadre des variétés ayant une régularité comme définie à la section 4.3.1 (page 64), à savoir : soit $f$ la fonction définie sur $I^s$ à valeurs dans $\{-1,+1\}$ dont nous cherchons l'approximation. La fonction $f$ possède la propriété de régularité suivante :
pour tout $x \in I^s$, il existe une boule $B(x_0,R)$ centrée en un point $x_0$ de rayon $R$ telle que :
- $x \in B(x_0,R)$ ;
- $f$ est constante sur $B(x_0,R)$.

Mathématiquement, cela se traduit par : $\exists R > 0$ tel que :
- $\forall x \in I^s, \exists x_0 \in I^s | x \in B(x_0,R)$ ;
- $\forall y \in B(x_0,R)\, f(y) = f(x)$.

Dans cette partie, nous supposons également que la dimension de la frontière de classification $\partial f$ est une sous-variété de dimension $s-1$, et nous supposons que la valeur $R$ de régularité est connue.

Dans une première partie, nous présentons la principe de l'algorithme avant de le définir. Dans une deuxième partie, nous démontrons la vitesse de convergence de l'erreur d'approximation. Nous concluons dans une troisième partie.

### 7.2.1 Présentation de l'algorithme

Notre algorithme se base sur le principe général de l'affinage d'une première grille de Shukarev $G_0$ de discrétisation de $[0,1]^s$ de résolution égale à $\frac{R}{2}$, *i.e.* les points successifs sur une dimension sont éloignés d'une distance de $\frac{R}{2}$. Notre algorithme, écrit en pseudo-code à l'algorithme 7.2.1, se base sur une partie d'exploration, puis sur une partie d'exploitation qui est itérée plusieurs fois.

**Partie exploration d'approximation des variétés :** les points de la grille initiale forment alors l'ensemble des points présentés à l'oracle par notre algorithme d'apprentissage actif. En choisissant une résolution de R/2, nous sommes sûrs de détecter toutes les zones de frontières entre les deux classes.

**Partie exploitation de l'approximation des variétés :** l'idée principale est d'affiner localement la grille d'apprentissage aux endroits où la fonction cible varie.
Nous définissons alors la notion de **sous-grille cubique minimale (SGCM)** pour désigner un ensemble de points de la grille formant les sommets d'un cube qui ne contient aucun autre point de la grille que ces sommets. On associe une résolution à chaque SGCM définie comme la longueur d'une arête du cube. Nous noterons $\mathscr{C}(G)$ l'ensemble des SGCM d'une grille $G$.
Nous définissons également l'ensemble $F_0$ des SGCM de la grille initiale qui possèdent des points ayant au moins un voisin de label différent du sien :

$$F_0 = \left\{ G_i \in \mathscr{C}(G_0) \text{ tel que } \exists (x_1,x_2) \in G_i^2 \text{ tels que } f(x_1) \neq f(x_2) \right\}.$$

Pour une SGCM notée $G$, nous notons $\mathscr{F}(G)$ l'ensemble des facettes du cube défini par $G$ sauf les sommets, *i.e.* l'"ensemble de toutes les facettes de dimension non-nulles, le cube lui-même compris.

Lorsque la fonction varie sur les sommets d'une sous-grille cubique minimale (**SGCM**), nous transformons cette SGCM et ses voisines en $2^s$ nouvelles SGCM de résolution la moitié de celle de la SGCM initiale, et nous présentons les nouveaux points à l'oracle. Nous itérons ensuite ce processus. Ce processus est la partie exploitation de l'algorithme.

**Conditions d'arrêt de l'algorithme d'apprentissage :** un nombre pré-défini d'itérations.

**Prédiction d'un nouveau point :** pour étiqueter un nouveau point, nous localisons la SGCM qui le contient. Si tous les sommets de cette SGCM sont de même label, alors nous lui attribuons ce label. Si les sommets de cette SGCM ne sont pas de même label, alors soit nous réalisons une prédiction aléatoire (avec une chance sur deux de se tromper), soit nous prenons, par exemple, le label du plus proche sommet du cube, soit nous relançons une nouvelle étape d'exploitation globale ou alors une exploitation uniquement localisée dans la SGCM qui contient le point et ses SGCM voisines.

**Explications des étapes d'exploitation de l'algorithme :** La figure 7.1 présente sur un exemple simple ces différents ensembles de points.

- à une étape $n$, l'ensembles $G_n$ est l'ensemble des points de la grille de $I^s$ et l'ensemble $F_n$ est l'ensemble des points de la grille qui disposent d'au moins un voisin de label différent ;
- nous identifions l'ensemble $\tilde{F}_n$ des SGCM qui sont soit non homogènes, soit sont voisines d'une SGCM non homogène ;
- nous divisons ces SGCM en ajoutant des points au milieu de chacune de ses facettes et nous obtenons l'ensemble $\tilde{F}_{n,\text{split}}$ ;
- nous présentons les nouveaux points créés à l'oracle ;
- nous remettons à jour la grille $G_{n+1}$ ainsi constituée ;
- nous identifions l'ensemble $F_{n+1}$ des SGCM de $\tilde{F}_{n,\text{split}}$ qui sont inhomogènes.

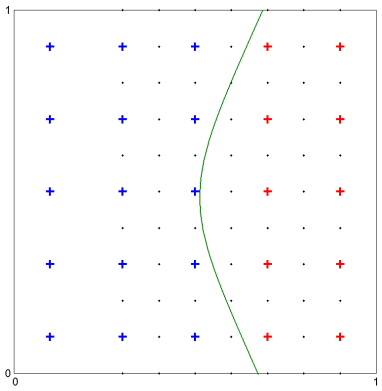



**FIGURE 7.1:** *Illustration des ensembles de points manipulés par l'algorithme d'exploration et d'exploitation des variétés. La grille de Shukarev $G_0$ initiale est représentée par les grosses croix rouges et bleues : les couleurs représentent les labels de ces points. La courbe verte représente la frontière de classification à estimer. L'ensemble $F_0$ correspond à l'ensemble des SGCM qui sont traversées par la frontière. L'ensemble $\tilde{F}_0$ est l'ensemble des SGCM traversées par la frontière ainsi leurs SGCM voisines : cela correspond alors à toutes les SGCM exemptées celles qui sont sur le côté gauche de la grille de Shukarev initiale. L'ensemble $\tilde{F}_{0,split}$ de nouveaux points à présenter à l'oracle, correspond alors à l'ensemble des points noirs de la figure.*

**Algorithme 7.2.1** : Un algorithme d'apprentissage actif pour approcher des variétés avec *a priori*.

- **Exploration :**
  - $G_0$ une grille de Shukarev de résolution $\frac{R}{2}$ ;
  - $F_0 = \left\{G_i \in \mathscr{C}(G_0) \text{ tel que } \exists (x_1,x_2) \in G_i^2 \text{ tels que } f(x_1) \neq f(x_2)\right\}$

- **pour** $n \geq 0$**, à partir de** $G_n$ **et** $F_n$ **:**
  **Exploitation :**
  1. $\tilde{F}_n = F_{n-1} \cup \left\{G_i \in \mathscr{C}(F_n) \text{ tel que } \exists G_j \in F_n \text{ tel que } \#(G_i \cap G_j) \geq 2\right\}$ ;
  2. $\tilde{F}_{n,\text{split}} = \bigcup_{G_i \in \tilde{F}_n} \bigcup_{S_k \in \mathscr{F}(G_i)} \text{milieu } (S_k)$ ;
  3. présenter à l'oracle les nouveaux points de $\tilde{F}_{n,\text{split}}$ ;
  4. $G_{n+1} = G_n \cup \tilde{F}_{n,\text{split}}$ ;
  5. $F_{n+1} = \left\{G_i \in \mathscr{C}(\tilde{F}_{n,\text{split}}) \text{ tel que } \exists (x_1,x_2) \in G_i^2 \text{ tel que } f(x_1) \neq f(x_2)\right\}$.

### 7.2.2 Étude de la convergence de l'algorithme

L'étape d'exploration consiste à utiliser une grille initiale de résolution inférieure à $\frac{R}{2}$ (où la valeur de $R$ est connue). Grâce aux hypothèses réalisées, toutes les composantes connexes des variétés sont alors connues et il s'avère alors juste nécessaire d'exploiter ces connaissances.

**Théorème.** *Constance de l'estimation sur des hypercubes homogènes à voisins homogènes*
*Pour toute sous-grille de configuration minimale G de la grille, si la fonction est homogène sur les sommets de G et sur les sommets des SGCM adjacentes de G, alors la fonction est homogène sur tout l'hypercube délimité par G.*

*Démonstration.* Nous réalisons la démonstration dans le cas de la grille initiale de Shukarev.
Supposons que le résultat du théorème soit faux et considérons que la grille $G$ contient un point $y$ de label positif alors que tous ces sommets sont de label négatif.
Avec l'hypothèse de régularité précédente, il existe un point $x$ de classe positive tel que

$$y \in B(x,R) \text{ et } B(x,R) \text{ est inclus dans la classe positive.}$$

Nous pouvons alors identifier trois cas de figures sur la position du point $x$ (voir la figure 7.2 pour une illustration graphique) :

- soit $x \in G$, alors, la SGCM $G$ de résolution $\frac{R}{2}$ est contenue dans cette boule de rayon $R$ de label négatif, et par conséquent $G$ est de label positif, ce qui est impossible par définition de $G$. Par conséquent, $x \notin G$ ;
- soit $x$ appartient à une grille $G'$ voisine et homogène avec $G$, alors l'intersection entre la boule et $G \cup G'$ n'est pas vide, *i.e.* $B(x,R) \cap G' \neq \emptyset$ : en effet, elle contient au mois un point de $G'$ car $G'$ est de résolution $R/2$ inférieure au rayon $R$ de la boule. Par conséquent, les points de cette intersection sont positifs, ce qui contredit l'homogénéité de $G'$ ;
- soit $x$ appartient à une grille $G''$ voisine d'une grille $G'$, $G'$ étant voisine et homogène à $G$, alors l'intersection entre la boule et $G'$ n'est pas vide, *i.e.* $B(x,R) \cap G' \neq \emptyset$ : en effet, elle contient au mois un point de $G'$. Par conséquent, les points de cette intersection sont positifs, ce qui contredit l'homogénéité de $G'$.

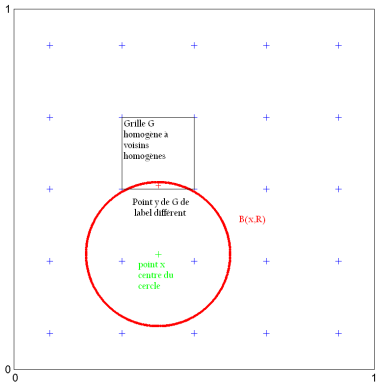



**FIGURE 7.2:** *Illustration pour démontrer qu'une grille homogène à voisins homogènes définie un hypercube constant : la SGCM G est homogène de label négatif (sommet de couleur bleue) et possède des voisins homogènes. S'il existe un point $y \in G$ de label positif (point de couleur rouge), alors par l'hypothèse de régularité, il existe un point x (de couleur verte) tel que la boule de rayon R et de centre x soit de label négatif. Cette boule contient nécessairement au moins un point soit de G, soit d'une SGCM homogène voisine de G : ce point est donc positif par appartenance à la boule : ce qui est absurde !*

Le point $x$ ne peut appartenir qu'à une de ces trois configurations. En effet, s'il n'appartient pas au voisinage d'ordre 2 de la grille $G$, alors avant d'intersecter la grille $G$, la boule $B(x,R)$ contient les grilles du voisinage d'ordre 2 de $G$ qui sont chacune de résolution $\frac{R}{2}$. Ainsi il ne peut intersecter la grille $G$, ce qui est absurde.

Nous venons de démontrer ainsi par l'absurde que toute SGCM homogène à voisinage homogène est constante sur l'hypercube qu'elle définit.
Ce résultat est démontré à la première itération de l'algorithme mais reste valable pour les autres itérations. En effet, le principe reste le même, et seule la localisation du point $x$ peut varier : l'ensemble des SGCM qui peuvent l'accueillir augmente. A l'itération $n$, le point $x$ peut appartenir à une des grilles du voisinage d'ordre $2^n$ de la grille considérée $G$. L'intersection entre la boule positive $B(x,R)$ et la grille $G$ et ses grilles voisines fournira une contradiction avec l'homogénéité de G ou des ses grilles voisines. □

**Théorème.** *Convergence de l'erreur en fonction du nombre de la complexité de l'échantillonnage. Sous la condition de régularité précédente, l'algorithme qui échantillonne les points et les étiquette comme défini précédemment, possède, après n itérations, un taux d'erreur en généralisation de l'ordre de*

$$\mathcal{O}\left(\frac{1}{2^n}\right), \tag{7.1}$$

*et a une complexité d'échantillonnage de l'ordre de*

$$\mathcal{O}\left(2^{n(s-1)}\right) \tag{7.2}$$

*points.*
*Par conséquent, le taux d'erreur est de l'ordre de* $\mathcal{O}\left(N^{-1/(s-1)}\right)$ *avec N le nombre de points présentés à l'oracle. La constante dans cet ordre dépend de la dimension et de la constante R dans l'hypothèse de régularité.*

*Démonstration.* Nous montrons que toutes les SGCM ne sont pas sélectionnées par la partie exploitation et estimons le nombre de points présentés à l'oracle. Par construction, cette partie de l'algorithme divise une grille $G$ à l'itération $n$ seulement s'il existe deux points de ses sommets qui sont de labels différents. Ces deux points sont nécessairement à une distance d'au plus $\frac{R\sqrt{s}}{2^n}$, *i.e.* distance maximale dans une grille obtenue sur une diagonale. La frontière entre les classes intersecte nécessairement le segment entre ces deux points. A partir d'une zone frontière de résolution $\frac{R}{2^n}$, et de largeur $4\frac{R}{2^n}$ (les grilles voisines), nous créons une bande de résolution $\frac{R}{2^{n+1}}$, et de largeur $4\frac{R}{2^{n+1}}$ autour de la frontière. Le nombre de SGCM à la frontière $\partial f$ est alors en $\mathscr{O}\left((2^n)^{s-1}\right)$. Le nombre de points lié à chaque bande $n$ est de l'ordre de $2^{n(s-1)}$. Ceci est réalisé pour l'itération $n$, en sommant sur toutes les valeurs de $n$, nous obtenons une équation similaire. Le coût total, pour les $n$ premières itérations est par conséquent en

$$N = \text{Nombres d'évaluations} = \mathscr{O}\left(2^{n(s-1)}\right). \tag{7.3}$$

Ce qui prouve l'équation 7.2.

Intéressons-nous maintenant à la précision de l'apprentissage à l'itération $n$ : par le lemme précédent, et pour $n$ suffisamment grand, pour toutes les SGCM ( *i.e.* toutes les sous-grilles de coté $\frac{R}{2^n}$) ont été subdivisées. Par conséquent, les grilles non divisées sont de classe constante, et l'erreur en généralisation du classifieur est nulle sur elles. Le taux d'erreur est alors borné supérieurement par

$$\underbrace{\mathscr{O}\left(2^{n(s-1)}\right)}_{\text{nombre de SGCM}} \times \underbrace{\frac{1}{2^{ns}}}_{\text{volume d'un hypercube défini par la grille}},$$

qui est $\varepsilon = \mathscr{O}\left(\frac{1}{2^n}\right)$ ; ce qui prouve l'équation 7.1.

L'équation 7.3 stipule que la précision est alors en $\varepsilon = \mathscr{O}\left(\frac{1}{N^{\frac{1}{s-1}}}\right)$. □

## 7.3 Conclusion

Dans ce chapitre, nous avons défini un algorithme de « batch active learning » à échantillonnage adaptatif utilisant l'hypothèse de régularité énoncée précédemment.

La connaissance de la valeur de la constante $R$ présente dans l'hypothèse de régularité permet de réaliser la partie exploration de l'algorithme en une seule étape, avec la garantie de ne rater aucune partie de la frontière. Les étapes suivantes consistent donc à ne réaliser que de l'exploitation.

Cet algorithme possède un taux d'erreur de l'ordre de $\left(\frac{1}{N}\right)^{\frac{1}{s-1}}$ après $N$ présentations d'instances à l'oracle ; ce qui est meilleur que les taux de $\left(\frac{1}{N}\right)^{1/s}$ et de $\left(\frac{\log(N)}{N}\right)^{1/(s-1+1/s)}$ présentés dans les travaux précédents. Notre définition de la régularité permet ainsi d'obtenir de meilleurs résultats, même si celle-ci peut paraître *a priori* plus contraignante que la définition utilisée dans les autres travaux. Néanmoins, notre définition de régularité engendre les fonctions de classification à dimension de Vapnik-Chervonenkis infinie, ce qui la rend utilisable pour de nombreux problèmes

difficiles. Cela correspond en fait à borner les dérivées pour garantir un rayon de courbure supérieur à une certaine valeur. Ce type de condition sur les dérivées locales de la variété est assez couramment utilisé.

Notre algorithme est réalisé à partir d'une grille que nous affinons localement. Cet affinage local ne nécessite alors qu'un nombre exponentiel de points avec une dimension en moins qu'une grille régulière. Un futur développement de cet algorithme consisterait à utiliser des suites à dispersion faible à la place d'une grille initiale, et de réduire ensuite la dispersion aux endroits où la fonction varie. Il faudrait alors déterminer une méthode de classification équivalente à celle qu'on utilise avec les SGCM. Utiliser des simplexes semble plus adéquat dans le cas où les points ne sont plus organisés sur une grille régulière. La détermination de ces simplexes peut être cependant un problème délicat...

Dans le cas où la constante $R$ n'est pas connue, il faut alors réaliser un pari sur sa valeur. A partir de ce pari, ou de son estimation, nous pouvons alors générer une grille de Shukarev de résolution R/2 et utiliser l'algorithme présenté dans ce présent chapitre. Cette estimation représente alors également une hypothèse sur le niveau de détails convenable que nous souhaitons obtenir sur l'approximation de la frontière de la variété : plus nous souhaitons une approximation de grande qualité, plus il faudra choisir initialement une valeur de $R$ petite ; et inversement ...

Enfin, une adaptation de cet algorithme au cas de classification multiclasse est naturellement et facilement envisageable.

# CONCLUSION

## De la thèse . . .

Lorsque j'explique ma problématique de thèse à des personnes non averties, je prends l'exemple d'un géologue confronté à l'analyse d'un terrain pour cartographier la présence de nappes d'or dans celui-ci à partir de $n$ prélèvements de terre à réaliser et à analyser. Les problèmes du géologue sont alors de deux types :

- où choisir les prélèvements à réaliser ?
- comment, à partir de cet échantillon prélevé, réaliser la cartographie ?

Les personnes ont alors souvent des yeux pétillants d'enthousiasme d'avoir pu comprendre un sujet de recherche de doctorat et de percevoir un intérêt concret à tout ce travail. Je leur explique alors que ce n'est pas la méthode utilisée en géologie, que les géologues ont des méthodes plus poussées mais que cette métaphore permet de mieux appréhender l'objet de mes recherches. Les enthousiasmes nés précédemment retombent alors aussi soudainement qu'ils sont apparus. . . J'enchéris que mes travaux ont cependant des intérêts scientifiques importants dans différents domaines tels que l'étude de systèmes ou l'intelligence artificielle. L'intérêt suscité précédemment retombe alors, [1] et je redeviens quelqu'un d'un peu fou qui essaye de faire de la science dans un labo. . . Je ne me suis encore jamais aventuré à expliquer qu'il peut exister différentes formes d'uniformité telles que la discrépance ou la dispersion, mais je suis persuadé que le résultat sur mes interlocuteurs serait équivalent !

Si je raconte de la même façon mon sujet de recherche à une personne avertie, celle-ci comprend alors que la deuxième question est un problème d'apprentissage et que la première question rend « actif » ce problème d'apprentissage.

**L'apprentissage sélectif diffère de l'apprentissage adaptatif :** dans cette thèse, nous nous sommes focalisés sur l'apprentissage actif pour l'approximation de variétés. La définition de l'apprentissage actif que nous utilisons n'est cependant pas la plus courante, et a pu paraître comme illégitime pour certains. Après tout, l'échantillonnage adaptatif n'est qu'un passage à la limite de l'échantillonnage sélectif ! Néanmoins, je suis persuadé que cette définition va prendre une part de plus en plus importante avec le temps et qu'il est alors utile de développer des méthodes d'échantillonnage adaptées. En effet, grâce aux développements des robots et des méthodes d'analyse automatisées, il sera de plus en plus courant d'inférer de l'intelligence à des machines pour qu'elles réalisent de manière autonome et satisfaisante des analyses selon un plan d'expérience qu'elles devront générer elles-mêmes. De même, avec l'augmentation significative des moyens de calculs, les scientifiques, toutes disciplines confondues, réalisent de plus en plus de modélisation, avec des modèles de plus en plus réalistes. Le besoin de méthodes d'analyse de ces modèles deviendra alors de plus en plus important.

1. Ce revirement d'intérêt peut aussi être attribué à mes qualités de pédagogue :p

Enfin, nous pouvons remarquer au cours des dernières années de plus en plus de workshop dédiés aux problèmes de coût d'étiquetage et plus globalement à l'apprentissage actif. A titre illustratif, John Shaw Taylor organise en fin d'année 2012 un workshop international [2] sur les coûts d'échantillonnage dans lequel l'échantillonnage adaptatif et l'ensemble de nos travaux peuvent avoir leurs places.

**L'apprentissage actif : un domaine à la croisée de deux disciplines :** nos travaux d'apprentissage actif se situent aussi à la croisée de deux disciplines scientifiques : l'apprentissage statistique et les « computer experiments », *i.e.* les plans d'expériences. Ces deux disciplines cohabitent mais n'interagissent pas.

Dans mes travaux de thèse, nous étions initialement confrontés à un problème d'apprentissage : comment générer théoriquement des points d'apprentissage ? Pour le résoudre, nous avons alors petit à petit dérivé vers un problème de computer experiments : comment générer en pratique ces points ? Nous avons alors mené de front ces deux axes de recherche en nous efforçant de réaliser un lien entre ces disciplines.

**Initialisation des points d'apprentissage, alias la naissance d'une problématique de recherche :** en mathématique, que ce soit pour la recherche de minima d'une fonction en optimisation ou pour générer des populations par exemple, il existe souvent une phase d'initialisation. L'apprentissage actif ne déroge pas à cette règle. Les premiers travaux consacrés à la « dérandomisation » des points d'apprentissage incitent à l'utilisation des suites à discrépance faible et sont dûs à Iwata et Ishii (2002) en classification et à Cervellera et Muselli (2003b) en régression. La phase d'initialisation de notre problème est alors résolue ;-)

Nous avons alors généré des suites à faible discrépance et des grilles régulières pour apprendre des problèmes de classification avec l'algorithme des SVMs. Ces suites qui, en théorie, donnent de bons résultats pour initialiser les points d'apprentissage, fournissent également des modèles plus parcimonieux (moins de vecteurs supports) : elles sont donc parfaitement adaptées à l'initialisation de points d'apprentissage. Cependant, leur utilisation délivre des modèles dont les capacités de généralisation sont inférieures aux modèles construits sur des grilles. Il s'agit peut-être d'une mauvaise implémentation informatique du problème par le thésard... Sauf qu'avec le temps et les implémentations, le problème demeure... J'obtenais ainsi un fait qui contredisait la théorie... Ayant accepté l'idée que la discrépance n'est pas le critère adéquat en classification [3], il a fallu petit à petit convaincre mon entourage, puis la communauté scientifique...

**Étude de ce nouveau problème de recherche :** à partir des résultats obtenus sur les expériences numériques, nous proposons la dispersion comme critère pour générer les premiers points d'apprentissage. En essayant de répondre à cette problématique de recherche, nous avons alors ouvert deux nouveaux fronts parallèles de recherche : i) prouver que la dispersion est bien un critère adéquat alors que nous ne disposons d'aucun élément théorique ; ii) savoir générer des suites à faible dispersion de taille quelconque.

Le premier front de recherche a conduit à cette présente thèse...

Le deuxième front de recherche apporte la plus-value algorithmique de cette thèse. Il est cependant intéressant de remarquer que cette notion de dispersion possède une caractéristique particulière : elle est à la fois ancienne et moderne, elle est présente dans de nombreux domaines scientifiques sous différentes dénominations mais qui s'ignorent. Elle apparaît dans les premiers

2. `https://sites.google.com/site/ieeecostsensitive/`
3. Inconsciemment, j'allais suivre la recommandation qu'un célèbre manufacturier clermontois faisait à ses ingénieurs et ouvriers : *Quand un fait contredit la théorie, retenons le fait !* (Édouard Michelin, 1924).

travaux de Niederreiter (1972) en optimisation. Elle réapparaît ensuite dans les « computer experiments » (voir Johnson *et al.* (1990)). Enfin, elle est réapparue l'année dernière (Fonteneau *et al.* (2011)) en apprentissage par renforcement sous le terme de sparsité !

## . . . aux perspectives

Dans cette deuxième partie de la conclusion, nous présentons différentes perspectives d'axes de recherche.

**Dispersion et malédiction de la dimension :** nous avons illustré le fait que le critère de faible dispersion résistait mieux à la malédiction de la dimension que celui de faible discrépance ; le comportement des suites aléatoires ne se rapproche pas de celui des grilles régulières. L'étude expérimentale a été réalisée sur des suites aléatoires de petite taille et de taille compatible avec les grilles de Shukarev. Il serait alors judicieux d'extrapoler ce résultat à des populations plus importantes de suites et aux suites de tailles quelconques.

**Dispersion, classification et malédiction de la dimension :** nous avons montré que minimiser la dispersion d'une suite est favorable par rapport aux suites aléatoires en terme de dispersion. Il serait alors judicieux d'extrapoler les simulations présentées au cours des chapitres précédents pour démontrer de manière plus significative que l'utilisation de la dispersion en classification ne souffre pas autant de la malédiction de la dimension que l'utilisation du critère de discrépance.

**Génération des suites à faible dispersion :** l'algorithme que nous proposons génère des suites dans l'hypercube unité. Dans la majorité des études, les données sont réduites et l'espace des individus est ramené à cet hypercube : toutes les variables jouent alors le même rôle. Dans certaines études, réaliser l'hypothèse de même rôle des variables n'est pas possible. Il serait alors préférable de savoir générer de telles suites dans des espaces non hypercubiques mais parallélépipédiques.

De façon similaire, il pourrait être intéressant de générer des suites à faible dispersion dans des espaces de forme quelconque, *i.e.* des espaces à fortes contraintes.

**Sélection initiale de points selon le critère de la dispersion :** dans le domaine industriel avec un contexte d'échantillonnage sélectif, il peut arriver qu'il y ait beaucoup d'instances candidates pour être présentées à l'oracle. Au CEA de Cadarache, les scientifiques sont confrontés à ce problème de choix des premiers points d'apprentissage pour analyser des surfaces de régression. Feuillard (2007) développa dans sa thèse des algorithmes pour sélectionner les points de la base d'instances en minimisant la discrépance obtenue. Nous pouvons très bien imaginer transférer ce problème à un problème de classification. Nous devrions alors sélectionner les instances minimisant la dispersion. Une question de recherche s'ouvre alors : comment sélectionner de manière intelligente dans une base de points ceux qui minimisent la dispersion ?

**Application à la sélection d'hyper-paramètres :** dans de nombreux problèmes d'apprentissage, et plus largement dans des problèmes de calibrage de modèles, la séléction des hyper-paramètres joue un rôle important dans la qualité du modèle. Bergstra et Bengio (2012) comparent cette sélection sur des valeurs issues d'une grille régulière classique et sur des valeurs issues d'une suite aléatoire. Il serait probablement intéressant d'étudier l'utilisation de ces suites à faible dispersion spour cette tâche. . .

**Exploration et exploitation dans un algorithme d'approximation de variétés :** notre algorithme pour exploiter une approximation de variété se base sur une grille que nous affinons localement. Une piste pour améliorer cet algorithme consisterait à remplacer la grille initiale par des suites à dispersion faible et les SGCM par des simplexes de taille décroissante. Nous obtiendrions probablement ainsi des approximations plus rapides des frontières des variétés.

**Apprentissage sur données bruitées :** nos études ont été menées sur des données sans présence de bruit. Il serait alors intéressant de regarder si nous obtenons les mêmes résultats en présence de données bruitées.

# ANNEXES

# - ANNEXE $\mathcal{A}$ -

# A PROPOS DE LA VARIATION D'UNE FONCTION AU SENS D'HARDY-KRAUSE

Dans cette annexe, nous présentons en détails la notion de variation d'Hardy-Krause d'une fonction et explicitons quelles sont les fonctions qui ont une variation finie au sens d'Hardy-Krause.

## A.1 Définition de la variation d'une fonction au sens d'Hardy-Krause

Soit une fonction $\varphi : I^s \to \mathbb{R}$ dont nous souhaitons estimer la variation d'Hardy-Krause.
Pour estimer cette variation, nous définirons d'abord la variation de Vitali, puis nous définirons celle d'Hardy-Krause. Enfin, nous nous intéresserons plus particulièrement au calcul de la variation des fonctions continues.

### A.1.1 Variation au sens de Vitali

Pour définir cette variation, nous commençons par « labeliser » et distinguer les sommets d'un intervalle élémentaire, puis calculons la variation de Vitali sur un intervalle élémentaire et, enfin, nous généralisons cette variation sur $I^s$. Ces trois étapes sont explicitées ci-après :

- **Distinction des sommets d'un intervalle élémentaire :** notons $B = \prod_{i=1}^{s} [a_i, b_i]$ un intervalle élémentaire de $I^s$. Pour chaque sommet, nous définissons un label binaire 0 à chaque point de type $a_i$ et un label 1 à chaque point de type $b_i$. Distinguons alors $e_B$ comme l'ensemble des sommets ayant un nombre pair de label 1 et $o_B$ comme l'ensemble des sommets ayant un nombre impair de label 1.

  **Exemple de distinction des sommets d'un intervalle :** considérons en dimension 2 l'intervalle $B = [x_1, x_2] \times [y_1, y_2]$. Le point $(x_2, y_2)$ est un point de type $b_i$ pour chaque dimension, par conséquent les labels associés sont 1 et 1, et il appartient donc à l'ensemble $e_B$. De façon identique, le point $(x_1, y_1)$ est un point de type $a_i$ pour chaque dimension, par conséquent les labels associés sont 0 et 0, et il appartient donc à l'ensemble $e_B$. Les points $(x_1, y_2)$ et $(x_2, y_1)$ sont deux points ayant chacun une coordonnée de type $a_i$ et une coordonnée de type $b_i$ : ils ont donc pour label 0 et 1, et appartiennent donc à l'ensemble $o_B$.

- **Variation de Vitali sur un intervalle élémentaire :** la variation de Vitali $\Delta(\varphi, B)$ sur l'intervalle élémentaire $B$ est la somme alternée des valeurs de la fonction $\varphi$ sur tous les sommets

de $B$ :

$$\Delta(\varphi, B) = \sum_{x \in e_B} \varphi(x) - \sum_{x \in o_B} \varphi(x).$$

– **Variation au sens de Vitali sur $I^s$ :** notons $\mathscr{P}$ une partition de $I^s$ en sous-intervalles. La variation de $\varphi$ sur $I^s$ au sens de Vitali est définie par :

$$V^{(s)}(\varphi) = \sup_{\mathscr{P}} \sum_{B \in \mathscr{P}} |\Delta(\varphi, B)|. \tag{A.1}$$

Enfin, pour $1 \leq k \leq s$ et $1 \leq i_1 < i_2 < \cdots < i_k \leq s$, notons $V^{(k)}(\varphi, i_1, \ldots, i_k)$ la variation au sens de Vitali de la restriction de $\varphi$ sur la face dimension $k$ : $\{(x_1, \ldots, x_s) \in I^s : x_i = 1 \text{ pour } i \neq i_1, \ldots, i_k\}$ où $x_i$ est la i$^e$ composante de $x$.

### A.1.2 Variation au sens d'Hardy-Krause

La variation de $\varphi$ sur $I^s$ au sens d'Hardy-Krause est définie par :

$$V_{HK}(\varphi) = \sum_{k=1}^{s} \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq s} V^{(k)}(\varphi, i_1, \ldots, i_k) \tag{A.2}$$

La variation au sens d'Hardy-Krause prend en compte l'ensemble des sous-directions de la base de l'espace. En effet, elle se compose de la somme de $2^s - 1$ variations de Vitalli. Ces variations pouvant être assez difficiles à calculer, la variation au sens d'Hardy-Krause peut être assez difficile à estimer, d'autant plus que la dimension augmente.

**Fonction à variation finie ou bornée :** une fonction $\varphi$ est dite à variation finie ou bornée si $V_{HK}(\varphi) < +\infty$.

### A.1.3 Calculs des variations dans le cas des fonctions continues

Nous avons vu précédemment que le calcul de la variation peut s'avérer être très difficile. Cette difficulté est due, d'une part au nombre d'estimations de la variation de Vitali, et d'autre part à la difficulté de ces estimations. Cependant, dans le cas des fonctions $\varphi$ à dérivées partielles continues, la variation au sens de Vitali s'écrit différemment et simplifie le calcul :

$$V^{(s)}(\varphi) = \int_0^1 \ldots \int_0^1 \left| \frac{\partial^s \varphi}{\partial x_1 \ldots x_s} \right| dx_1 \ldots dx_s. \tag{A.3}$$

Le calcul de la variation au sens d'Hardy-Krause reste inchangé.

## A.2 Quelle classe de fonctions possède une variation d'Hardy-Krause finie ?

L'hypothèse d'une variation d'Hardy-Krause finie est une hypothèse très forte et non naturelle. Les trois exemples ci-après illustrent cette difficulté :

1. Supposons les fonctions $\varphi_1(t) = \prod_{i=1}^{s}(1 - t_i)$ et $\varphi_2(t) = \prod_{i=1}^{s} t_i$ définies sur $I^s$. Ces deux fonctions sont symétriques et de mêmes intégrales. Néanmoins, elles ne possèdent pas la même variation finie au sens d'Hardy-Krause. En effet, nous avons alors $V_{HK}(\varphi_1) = 1$ et $V_{HK}(\varphi_2) = 2^s - 1$.

2. Si $\varphi_1$ et $\varphi_2$ sont deux fonctions linéaires de l'hyper-cube de dimension $s$, alors $\min(f,g)$ est une fonction à variation finie au sens d'Hardy-Krause lorsque $s=2$, mais pas nécessairement lorsque $s>2$.

3. Les fonctions indicatrices : une fonction indicatrice en dimension 2 est de variation d'Hardy-Krause finie si, soit ses variations Vitali sont positives, soit elle n'est pas dépendante d'au moins une de ses variables d'entrées. Cette règle n'est plus vraie lorsque $s \geq 3$.

D'une manière générale, lorsque la dimension augmente, il est de plus en plus difficile pour une fonction d'être à variation finie.

**Classe de fonction à variation finie :** les fonctions dont toutes les dérivées partielles sont des fonctions continues à variation finie possèdent une variation d'Hardy-Krause finie. Ceci est équivalent à la classe des fonctions $\mathscr{C}^s$. Cette condition est une condition suffisante et non nécessaire.

**Propriétés de la variation d'Hardy-Krause :** si $\varphi_1$ et $\varphi_2$ sont deux fonctions à variation finie d'Hardy-Krause alors les fonctions $\varphi_1 + \lambda\varphi_2$ où $\lambda \in \mathbb{R}$, et $\varphi_1 \times \varphi_2$ sont aussi à variation finie d'Hardy-Krause. De plus si la fonction $\varphi_1$ et de variation finie d'Hardy-Krause, et est bornée inférieurement en module par un réel positif non nul, alors la fonction $\dfrac{1}{\varphi_1}$ est aussi à variation finie.

**Qu'apporte l'hypothèse de variation finie d'une fonction ?** Cette hypothèse implique qu'une fonction à faible variation au sens d'Hardy-Krause sera « régulière » et ne variera pas beaucoup. Néanmoins, la réciproque est fausse : une fonction peu évolutive n'aura pas nécessairement une variation faible : le premier exemple de ce début de sous-section illustre bien ces propos. Cette variation n'apporte aucune assurance que des propriétés de régularité de la fonction soient bien exprim(ées.

# - ANNEXE $\mathcal{B}$ -

## RÈGLES DE CLASSIFICATION GÉNÉRÉES

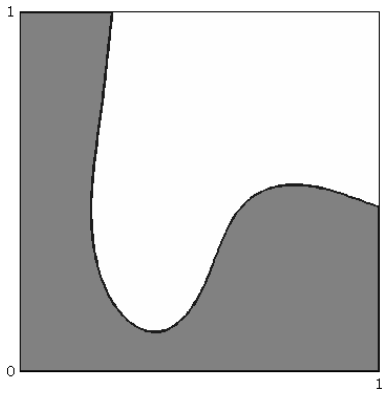
La génération de points n'est pas une problématique habituelle dans la comparaison des méthodes d'apprentissage (en apprentissage actif ou non). Par conséquent, il n'existe pas de bases de référence de règles de classification dans la base de données du *UCI - Center for Machine Learning and Intelligent Systems* de l'Université de Californie (voir Frank et Asuncion (2010)).
En classification, la règle utilisée en littérature est l'apprentissage d'une boule. Cette règle est très simple et n'est pas optimale dans les étapes d'exploitation des modèles.
Une autre règle utilisable est d'apprendre deux spirales imbriquées : cette règle est difficile et possède d'importantes variations : elle est surtout utilisée pour comparer des méthodes d'apprentissage plutôt que pour développer des méthodes.
Afin de développer nos méthodes, nous aurions pu faire le choix de l'étude d'un modèle réel. Cependant, l'étiquetage des points dans les modèles réels se fait souvent grâce à des simulations et nécessite un temps de réponse non anodin à chaque sollicitation. Afin de limiter les durées des expériences numériques, nous avons choisi de générer nos propres règles de classification.

Nous avons généré des règles de classification avec des difficultés différentes, *i.e.* des régularités et/ou composantes connexes différentes. Dans chaque ensemble de règles générés, nous avons une proportion d'un tiers de règles « simples ».

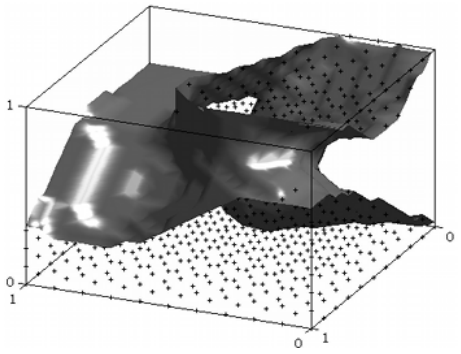
Les règles de classification simples se caractérisent par des frontières lisses soumises à peu de variation. Ces règles satisfont, en général, l'hypothèse de régularité du théorème présenté à la section 4.3.1 à la page 64. Des représentations graphiques de ces règles sont données en dimension 2 à la figure B.1(a) et en dimension 3 à la figure B.1(b).

Les règles de classification difficiles se caractérisent par des frontières avec des variations moins lisses et plus importantes. De plus, les surfaces de classifications possèdent plus de composantes connexes. Des représentations graphiques de ces règles sont présentées en dimension 2 à la figure B.1(a) et en dimension 3 à la figure B.2(b).

Explicitement, nous avons utilisé la procédure suivante pour générer des règles de classification : nous avons généré dans l'hyper-cube unité $k_1$ points de référence étiquetés $+1$ et $k_2$ points de référence étiquetés $-1$. Soit $k \in \mathbb{N}$ tel que $k \leq \min(k_1, k_2)$. Pour étiqueter un nouveau point $x$, nous calculons la distance moyenne $d_1$ entre $x$ et les $k$ plus proches points positifs de référence ainsi que la distance $d_2$ entre $x$ et les $k$ plus proches points négatifs de référence. Si $d_1$ est supérieure à $d_2$, le point $x$ est étiqueté $+1$, sinon il est étiqueté $-1$. Les valeurs relatives de $k_1, k_2$ et $k$ changent la complexité des règles.

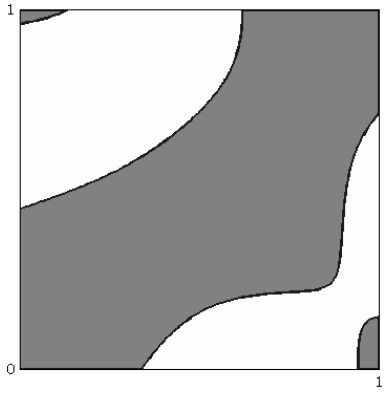



(a) Dimension 2

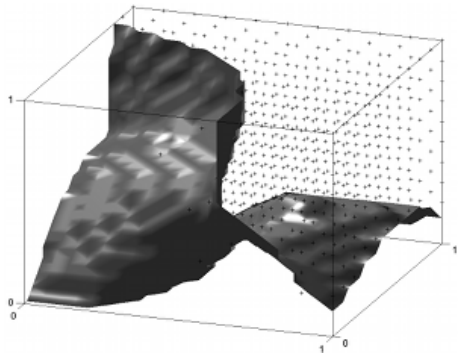



(b) Dimension 3

**FIGURE B.1:** *Exemples de règles de classification simples. Pour des raisons de clarté, seuls les points d'une classe sont représentés.*




(a) Dimension 2




(b) Dimension 3

**FIGURE B.2:** *Exemples de règles de classification difficiles. Ces règles possèdent plusieurs composantes connexes et ont une frontière peu lisse. Pour des raisons de clarté, seuls les points d'une classe sont représentés.*

# - ANNEXE C -

# MÉTHODES DE RÉDUCTION DE LA VARIANCE D'ESTIMATION D'INTÉGRALES PAR LA MÉTHODE DE MONTE-CARLO

Dans cette annexe, nous présentons les différentes méthodes usuelles de réduction de l'erreur d'estimation d'intégrales, notamment les méthodes de réduction de la variance de l'estimation, par la méthode de Monte-Carlo avec un point de vue probabiliste.

## C.1 Estimation d'intégrales par la méthode de Monte-Carlo

Supposons que l'on souhaite calculer l'intégrale d'une fonction $g$ sur un domaine $D$ donné, *i.e.* $I = \int_D g(x)\,dx$. Ce calcul peut être difficile. En considérant une variable aléatoire $X$ uniforme sur $D$ , on peut voir alors l'intégrale $I$ comme étant l'espérance de la variable aléatoire $g(X)$ sur $D$ : *i.e.* $I = \mathbb{E}[g(X)]$. Un estimateur de cette espérance, et donc de cette intégrale, est la moyenne de réalisations de la variable aléatoire $X$, qui est facile à obtenir. Cette méthode, dite de Monte-Carlo, est explicitée au chapitre 2.1 (page 26)

D'une manière générale, soient X une variable aléatoire de fonction de densité $f_X(x)$ et $g$ une fonction donnée. Nous souhaitons calculer l'espérance $\mathbb{E}[g(X)]$ de la variable aléatoire $g(X)$ : il est alors très probable qu'il soit difficile d'obtenir une valeur de celle-ci de manière « classique », notamment pour connaître sa fonction de densité . Cependant, il est possible de l'estimer par la méthode de Monte-Carlo. Pour résumer, celle-ci consiste à générer $N$ réalisations indépendantes $X_i$ de la variable $X$ et de les utiliser pour évaluer l'estimateur. L'espérance $\mathbb{E}[g(X)]$ est approximée par la moyenne arithmétique : $\hat{h}(X) = \frac{1}{N}\sum_{i=1}^{N} g(X_i)$.

Un des problèmes de cette technique est d'obtenir une estimation de la qualité de l'estimation de l'intégrale en fonction du nombre de points utilisés. Pour cela, nous présentons deux méthodes probabilistes apportant des réponses à ce problème : la méthode de la loi forte des grands nombres et la méthode du théorème central limite.

### C.1.1 Estimation de qualité par la loi forte des grands nombres

D'après la loi forte des grands nombres, $\forall \varepsilon > 0, \mathbb{P}\left[\left|h - \hat{h}\right| < \varepsilon\sigma_h\right] \geq 1 - \frac{1}{N\varepsilon^2}$. Nous pouvons choisir un degré de confiance $\alpha$ de façon à construire un intervalle de confiance de $h$ avec une confiance de $\alpha$. Ainsi $\Pr\left[h \in \left[\hat{h} - \varepsilon\sigma_h, \hat{h} + \varepsilon\sigma_h\right]\right] \geq 1 - \alpha$. Ceci implique alors que $N > \frac{1}{\alpha\varepsilon^2}$.

En pratique, $\sigma_h$ n'est pas connu et nous pouvons l'estimer empiriquement par l'estimateur $\hat{\sigma_h^2} = \frac{1}{N-1}\sum_{i=1}^{N}(g(X_i)-\hat{h})^2$.

### C.1.2 Estimation de la qualité par le théorème central limite

Cette technique permet d'améliorer sensiblement le nombre de simulations nécessaires.
Définissons $Z = \frac{1}{\sigma_h\sqrt{N}}\sum_{i=1}^{N}(g(X_i)-h) = \frac{\sqrt{N}}{\sigma_h}\sum_{i=1}^{N}(g(X_i)-h) = \frac{\sqrt{N}}{\sigma_h}(\hat{h}-h)$. La variable aléatoire ainsi formée est de moyenne nulle et de variance unitaire.
De plus, nous avons les relations suivantes :

$$\mathbb{P}\left[|\hat{h}-h| < \varepsilon\sigma_h\right] = \mathbb{P}\left[\frac{\sqrt{N}}{\sigma_h}|\hat{h}-h| < \frac{\sqrt{N}}{\sigma_h}\varepsilon\sigma_h\right] = \mathbb{P}\left[|Z| < \varepsilon\sqrt{N}\right].$$

Lorsque $N$ est grand, par le théorème de la limite centrale, la variable aléatoire $Z$ s'approxime par une loi gaussienne centrée réduite. D'où :

$$\begin{aligned}\mathbb{P}\left[|Z| < \varepsilon\sqrt{N}\right] &\approx \frac{1}{\sqrt{2\pi}}\int_{-\varepsilon\sqrt{N}}^{\varepsilon\sqrt{N}} e^{-\frac{x^2}{2}}dx \\ &= \frac{2}{\sqrt{\pi}}\int_0^{\varepsilon\sqrt{\frac{N}{2}}} e^{-u^2}du \\ &= erf(\varepsilon\sqrt{\frac{N}{2}})\end{aligned}$$

en notant $erf$ la fonction de répartition d'une loi normale centrée réduite.
Si nous désirons exprimer le nombre de simulations nécessaires en fonction de la confiance $\alpha$ de l'estimation, alors $\mathbb{P}\left[|Z| < \varepsilon\sqrt{N}\right] > 1-\alpha = erf(\beta)$ où $\beta = erf^{-1}(1-\alpha)$.
Ceci implique $erf(\varepsilon\sqrt{\frac{N}{2}}) \geq erf(\beta)$, et comme la fonction $erf$ est croissante :
$\varepsilon\sqrt{\frac{N}{2}} \geq (\beta)$, d'où $N \geq 2\left(\frac{\beta}{\varepsilon}\right)^2$.

## C.2 Méthodes d'amélioration des estimations

Dans la section précédente, nous avons vu que pour augmenter la précision de l'estimation de l'intégrale, il est nécessaire d'augmenter le nombre de simulations. Cependant, la simulation d'une expérience peut s'avérer coûteuse en temps, il est donc nécessaire de réduire le nombre de simulations.

Dans cette partie, nous nous intéressons à différentes techniques de réduction de temps de calcul, i.e. du nombre de simulations. Ces méthodes sont la méthode des variables antithétiques, des variables de contrôle, la méthode d'« Importance Stratification » et la méthode de stratification.

### C.2.1 Méthode des variables antithétiques

Cette méthode permet de diviser par deux le nombre de simulations nécessaires en se basant sur le principe suivant : pour évaluer la moyenne d'une variable aléatoire à faible variance, il faut peu d'individus, pour une variable à forte variance, il faut beaucoup d'individus.

La mise en pratique de cette idée consiste à générer $N$ variables aléatoires $X_i$ et à construire $N$ autres variables $X_i^a$ de même loi de probabilité et corrélées négativement avec celles-ci. La variance de la moyenne empirique obtenue sur les $N$ variables est alors plus faible que celle obtenue avec $2N$ variables indépendantes.

*Démonstration.* Posons $\hat{m}_a = \frac{1}{2N}\sum_{k=1}^{N}(X_k + X_k^a)$ où $\mathbb{E}\left[(X_k^a - m_X)(X_j - m_X)\right] = -\delta_{k,j}\sigma_X^2$.
Il est trivial de voir que nous obtenons bien la même espérance. Concernant la variance :

$$\begin{aligned}
\mathbb{V}ar(\hat{m}_a) &= \mathbb{E}\left[(\hat{m}_a - m_X)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{2N}\sum_{k=1}^{N}\left((X_k + X_k^a) - m_X\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{2N}\sum_{k=1}^{N}X_k + \frac{1}{2N}\sum_{k=1}^{N}X_k^a - m_X\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{2N}\sum_{k=1}^{N}(X_k - m_X) + \frac{1}{2N}\sum_{k=1}^{N}(X_k^a - m_X)\right)^2\right] \\
&= \frac{1}{4N^2}\sum_{k=1}^{N}\sigma_X^2 + \frac{1}{4N^2}\sum_{k=1}^{N}\sigma_X^2 + \frac{1}{4N^2}\sum_{k=1}^{N}\sum_{j=1}^{N}\mathbb{E}\left[(X_k^a - m_X)(X_j - m_X)\right] \\
&= \frac{1}{2N}\sigma_X^2 - \frac{1}{4N}\sigma_X^2 \\
&= \frac{1}{4N}\sigma_X^2.
\end{aligned}$$

Dans le cadre de variables indépendantes, la variance empirique de $2N$ variables est égale à $\frac{1}{2N}\sigma_X^2$. □

En théorie, cette méthode permet donc d'améliorer, avec un facteur deux, la qualité de la variance de l'estimation. Néanmoins, elle suppose que l'on soit capable de générer des variables parfaitement corrélées négativement, une corrélation partielle pouvant augmenter l'incertitude du résultat…

### C.2.2 Méthode des variables de contrôle

Cette méthode se base sur le même principe que la précédente, *i.e.* introduire des variables corrélées négativement avec les variables initiales pour réduire la variance de l'estimateur. Néanmoins, aucune hypothèse sur la distribution de ces variables n'est faîte.

**Méthode avec une seule variable de contrôle :** soient $X$ une variable aléatoire et $Y$ une variable aléatoire corrélée négativement avec $X$ et de moyenne connue. $Y$ est alors appelée la variable de contrôle.
Définissons la variable aléatoire $Z = X + \alpha(Y - \mathbb{E}[Y])$. Cette variable possède ainsi la même moyenne que $X$, *i.e.* $m_X = m_Z$. L'objectif consiste donc à choisir $\alpha$ de manière à réduire la variance de $Z$.

Cette variance de $Z$ s'exprime ainsi :

$$\begin{aligned}\mathbb{V}ar(Z) &= \mathbb{E}\left[(Z-m_Z)^2\right] \\ &= \mathbb{E}\left[(X-\alpha(Y-\mathbb{E}[Y])-m_Z)^2\right] \\ &= \mathbb{V}ar(X)+\alpha^2\mathbb{V}ar(Y)+2\alpha Cov(X,Y) \text{ où } Cov(X,Y)=\mathbb{E}\left[(X-m_x)(Y-m_Y)\right].\end{aligned}$$

Le choix optimal $\alpha^*$ est tel que : $\dfrac{d\mathbb{V}ar(Z)}{d\alpha}=0 \Rightarrow 2\alpha^*\mathbb{V}ar(Y)+2Cov(X,Y)=0$, d'où $\alpha^*=-\frac{Cov(X,Y)}{\mathbb{V}ar(Y)}$.

Nous obtenons alors $\mathbb{V}ar(Z)=\mathbb{V}ar(X)-\dfrac{[Cov(X,Y)]^2}{\mathbb{V}ar(Y)}$, et la variance de $Z$ est plus faible que celle de $X$.

**Généralisation avec plusieurs variables de contrôle :** nous pouvons utiliser plusieurs variables de contrôle $Y_1, Y_2, \ldots, Y_k$ de $X$ et écrire $Z$ sous la forme :

$$Z = X+\alpha_1(Y_1-\mathbb{E}[Y_1])+\cdots+\alpha_k(Y_k-\mathbb{E}[Y_k]) = X+\underline{\alpha}^T(\underline{Y}-\underline{m_Y})$$

en posant $\underline{\alpha}=(\alpha_1,\ldots,\alpha_k)^T$, $\underline{Y}=-(Y_1,\ldots,Y_k)^T$ et $\underline{m_Y}=(\mathbb{E}[Y_1],\ldots,\mathbb{E}[Y_k])^T$.
Nous obtenons alors $\mathbb{V}ar(Z)=\mathbb{E}\left[(X-m_x+\underline{\alpha}^T(\underline{Y}-\underline{m_Y})(X-m_x+(\underline{Y}-\underline{m_Y})^T\underline{\alpha}\right]$.
D'où $Var(Z)=\mathbb{V}ar\binom{X}{}=\underline{\alpha}^T\underline{\Lambda_\alpha}\underline{\alpha}+2\underline{\alpha}^T\underline{P}$,
en notant $\underline{P}=\mathbb{E}[(\underline{Y}-\underline{m_Y})(\underline{X}-m_X)]=(Cov(X,Y_1),\ldots,Cov(X,Y_k))^T$.
En supposant que la matrice de variance-covariance de $\underline{Y}$ est définie positive (donc inversible), le vecteur optimal $\underline{\alpha}^*$ correspond à la solution de l'équation :

$$\frac{dVar(Z)}{d\underline{\alpha}}=\left(\frac{dVar(Z)}{d\alpha_1},\ldots,\frac{dVar(Z)}{d\alpha_k}\right)^T=2\underline{P}+2\underline{\Lambda_Y}\underline{\alpha}=0.$$

Soit $\underline{\alpha}^*=-\underline{\Lambda_Y}^{-1}\underline{P}$. La variance de $Z$ s'écrit alors $Var(Z)=\mathbb{V}ar(X)-\underline{P}^T\underline{\Lambda_Y}^{-1}\underline{P}$. Le deuxième terme est positif car il représente une forme quadratique positive et $\underline{\Lambda_Y}^{-1}$ est définie positive. D'où le résultat.

### C.2.3 Méthode d'échantillonnage préférentiel ou « Importance Sampling »

Dans les méthodes précédentes, nous utilisons la moyenne empirique comme estimateur. La précision statistique est alors inversement proportionnelle à la variance de la variable aléatoire dont on veut estimer la moyenne. Dans le cadre d'événements rares, *i.e.* pour une variance faible, il est souvent nécessaire d'avoir un nombre énorme de simulations pour une précision demandée. Dans cette situation, les méthodes précédentes sont inadaptées.

Dans cette section, nous présentons une nouvelle méthode moins sensible à ce problème : la méthode de l'échantillonnage préférentiel, ou *« importance sampling »* en anglais. Cette méthode consiste simplement à réaliser un changement de variables pour réduire la variance de la variable à simuler.

Soit $X$ une variable aléatoire de densité de probabilité $f_X$, et $Y$ la variable aléatoire obtenue par une transformation de $X$, *i.e.* $Y=h(X)$. Nous pouvons montrer alors que $m_y=\mathbb{E}[Y]=\int h(x)\,f_X(x)\,dx$, et pour $N$ échantillons $X_i$, la moyenne empirique est égale à $\hat{m}_Y=\hat{\mathbb{E}}[Y]=\dfrac{1}{N}\sum_{i=1}^{N}h(X_k)$ où les échantillons $X_i$ sont indépendants et de même fonction de densité $f_X$.
Soit $g_Y$ une fonction de densité de probabilité quelconque.

En posant $Z(y) = h(y)\frac{f_X(y)}{g_Y(y)}$, alors $\mathbb{E}[Y] = \int h(y)\frac{f_X(y)}{g_Y(y)}g_Y(y)\,dy = \int Z(y)\,g_Y(y)\,dy$. $Z$ est une variable aléatoire, obtenue par transformation de la variable aléatoire $Y$, et de densité $g_Y$.
La moyenne empirique de $Z$ est alors donnée par $\hat{m}_Z = \hat{\mathbb{E}}[Z] = \frac{1}{N^*}\sum_{i=1}^{N^*} Z(Y_k)$, où les $N^*$ variables aléatoires $Y_k$ sont indépendantes et de même loi de densité $g_Y$. Nous avons alors $\mathbb{E}\left[\hat{\mathbb{E}}[Y]\right] = \mathbb{E}\left[\hat{\mathbb{E}}[Z]\right]$. Nous pouvons donc utiliser $\hat{\mathbb{E}}[Y]$ pour estimer $\mathbb{E}[Z]$. La précision de $\hat{\mathbb{E}}[Z]$, quand $N$ et $N^*$ sont grands, est donnée par $N^* \geq Q^{-1}(\frac{\eta}{2})\frac{1}{k^2}\frac{Var(Z)}{\mathbb{E}[Z]^2}$ où $Q(x) = \frac{1}{2}ercf\left(\frac{x}{\sqrt{2}}\right)$. Nous avons donc $N^*$ qui dépend de $\mathbb{V}ar(Z)$ : pour réduire $N^*$, il faut réduire $\mathbb{V}ar(Z)$.
Le problème à résoudre consiste donc à trouver $g_Y$ permettant de réduire sensiblement $\mathbb{V}ar(Z)$ ...

Lorsque la fonction $g$ est multi-dimensionnelle, nous pouvons procéder, en général, de manière indépendante pour chaque dimension.

### C.2.4 Méthode de stratification

Cette méthode est issue de la théorie des sondages. Le principe est de découper la zone d'intégration en sous-zones et d'affecter davantage de tirages aux zones aux plus grandes variances. Nous cherchons toujours à estimer $I = \mathbb{E}[g(X)] = \int g(x)f(x)\,dx$ où $X$ est un vecteur aléatoire de $s$ dimensions et de densité $f$.

Supposons que l'espace d'intégration soit $\mathbb{R}^s$ et que $(D_i, i = 1,\ldots,m)$ en soit une partition. Alors $I = \sum_{i=1}^{m}\mathbb{E}[g(X)|X \in D_i]\,\mathbb{P}[X \in D_i]$.
La proportion de $I$ relative à $D_i$ est alors $I_i = \frac{\mathbb{E}[\mathbb{1}_{\mathbb{X}\in\mathbb{D}}(\mathbb{X})]}{\mathbb{P}[X \in D_i]}$.
Supposons connues les probabilités $p_i = \mathbb{P}[X \in D_i]$. Nous pouvons alors redéfinir les variables $X^i$ comme une variable aléatoire suivant la loi de $X$ restreinte à $D_i$. Nous simulons $X^i$ par une quelconque méthode (par rejet par exemple), puis nous estimons $I_i$ en utilisant un échantillon de taille $n_i$ par $\hat{I}_i = \frac{1}{n_i}\left(X_1^i + \cdots + x_{n_i}^i\right)$ et $I$ par $\hat{I} = \sum_{i=1}^{m}\mathbb{P}[X \in D_i]\hat{I}_i$.
La méthode converge lorsque tous les $n_i$ tendent vers l'infini. Le problème consiste donc à choisir les $n_i$ avec pour contrainte $\sum_{i=1}^{m} n_i = n$.
Nous devons donc minimiser $\mathbb{V}ar\left(\hat{I}\right) = \sum_{i=1}^{m} {p_i}^2\mathbb{V}ar\left(\hat{I}_i\right) = \sum_{i=1}^{m} {p_i}^2\frac{\sigma_i}{n_i}$.
Par la technique des multiplicateurs de Lagrange , on obtient $n_i = n\frac{p_i\sigma_i}{\sum_{i=1}^{m} p_i\sigma_i}$.

Lorsque le partitionnement est adéquat, nous observons alors une réduction de la variance. Le choix de la partition est donc délicat, sauf lorsque les variables sont catégorisées (sexe, géographie, couleur, ...).

### C.2.5 Méthode géométrique

Des méthodes géométriques de réduction existent. Nous ne les présentons pas en détail car celles-ci sont peu utilisées en pratique. Nous pouvons citer par exemple Munos (2005) qui utilise alors des chaînes de Markov.

# BIBLIOGRAPHIE


ABE, N. et MAMITSUKA, H. (1998). Query Learning Strategies Using Boosting and Bagging. *In ICML'98 : Proceedings of the Fifteenth International Conference on Machine Learning*, pages 1–9, San Francisco, CA (USA). Morgan Kaufmann Publishers Inc.

ANGLUIN, D. (1988). Queries and concept learning. *Machine Learning*, 2:319–342.

ANGLUIN, D. (2001). Queries revisited. *In* ABE, N., KHARDON, R. et ZEUGMANN, T., éditeurs : *Algorithmic Learning Theory*, volume 2225 de *Lecture Notes in Computer Science*, pages 12–31. Springer Berlin / Heidelberg.

ANGLUIN, D. et LAIRD, P. (1988). Learning from noisy examples. *Machine Learning*, 2:343–370.

AUBIN, J. (1991). *Viability theory*. Birkhäuser.

AUFFRAY, Y., BARBILLON, P. et MARIN, J.-M. (2012). Maximin design on non hypercube domain and kernel interpolation. *Statistics and Computing*, 22(3):703–712.

BALCAN, M.-F., BEYGELZIMER, A. et LANGFORD, J. (2006). Agnostic active learning. *In ICML'06 : Proceedings of the International Conference on Machine Learning*, pages 65–72.

BALCAN, M.-F., HANNEKE, S. et WORTMAN, J. (2008). The true sample complexity of active learning. *In COLT'08 : Proceedings of the Conference on Computational Learning Theory*, pages 45–56.

BALCÁZAR, J., CASTRO, J. et GUIJARRO, D. (2001). A general dimension for exact learning. *In COLT'01 : Proceedings of the 14th ACM Conference on Computational Learning Theory*, pages 354–367. Springer.

BALCÁZAR, J. L., CASTRO, J., GUIJARRO, D., KOBLER, J. et LINDNER, W. (2007). A general dimension for query learning. *Journal of Computer and Sytem Sciences*, 73:924–940.

BELLMAN, R. (1961). *Adaptative Control Processes*. Princeton University Press.

BERGSTRA, J. et BENGIO, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

BLUM, A. et MITCHELL, T. (1998). Combining labeled and unlabeled data with co-training. *In COLT'98 : Proceedings of the eleventh annual conference on Computational Learning Theory*, pages 92–100, New York, NY (USA). ACM.

BONDU, A. (2008). *Apprentissage Actif par Modèles Locaux*. Thèse de doctorat, Université d'Angers.

BONDU, A. et LEMAIRE, V. (2007a). Active learning using adaptive curiosity. *In ICER'07 : Proceedings of International Conference on Epigenetic Robotics, Piscataway, New Jersey (USA)*.

BONDU, A. et LEMAIRE, V. (2007b). État de l'art sur les méthodes statistiques d'apprentissage actif. *RNTI : Revue des Nouvelles Technologies de l'Information*.

BONDU, A., LEMAIRE, V. et BOULLÉ AND, M. (2010). Exploration vs. exploitation in active learning : A bayesian approach. *In IJCNN'10 : Proceedings of the the 2010 International Joint Conference on Neural Networks*, pages 1 –7.

BONNEUIL, N. (2003). Making ecosystem models viable. *Bulletin of Mathematical Biology*, 65(6): 1081–1094.

BOULLÉ, M. (2006). MODL : A bayes optimal discretization method for continuous attribute. *Machine Learning*, 65:131–165.

BREIMAN (1993). *Classification and Regression Trees*. Chapman & Hall.

BUNDSCHUH, P. et ZHU, Y. (1993). A method for exact calculation of the discrepancy of low-dimensional finite point sets. *Abhandlungen aus dem Mathematischen Seminar der Universite Hamburg (Allemagne)*, 63:115–133. 10.1007/BF02941337.

BURNASHEV, M. et ZIGANGIROV, K. (1974). An interval estimation problem for controlled observations. *Problems in Information Transmission*, 10:223–231.

CAMPBELL, C., CRISTIANINI, N. et SMOLA, A. (2000). Query learning with large margin classifiers. pages 111–118. Morgan Kaufmann.

CANTU PAZ, E. et KAMATH, C. (2003). Inducing oblique decision trees with evolutionary algrothms. *IEEE Transactions on Evolutionary Computation*, 7:54–69.

CASTRO, R., WILLETT, R. et NOWAK, R. (2005). Faster rates in regression via active learning. *In NIPS'05 : Proceedings of Neural Information Processing Systems*, pages 179–186.

CASTRO, R. M. et NOWAK, R. D. (2007). Minimax bounds for active learning. *In COLT'07 : Proceedings of the Conference on Computational Learning Theory*, pages 151–156. Verlag.

CERVELLERA, C., MACCIO, M. et MUSELLI, M. (2008). Deterministic Learning for Maximum-Likelihood Estimation Through Neural Network Learning. *In Proceedings IEEE Transactions on Neural Network*, volume 19(8), pages 1456–1467.

CERVELLERA, C. et MUSELLI, M. (2003a). A Deterministic Learning Approach Based on Discrepancy. *In* B. APOLLONI, M. Marinaro, R. T., éditeur : *Proceedings of Neural Nets : WIRN VIETRI '03, vol. 2859 of Lecture Notes in Computer Science*, pages 53–60. Berlin : Springer-Verlag.

CERVELLERA, C. et MUSELLI, M. (2003b). Pattern Recognition as a Deterministic Problem : An approach based on discrepancy. *In Proceedings IEEE Artificial Neural Networks in Pattern Recognition - Proceedings of the First IAPR-TC3 Workshop, Florence (Italy), 5-7 June 2003*, pages 139–145.

CERVELLERA, C. et MUSELLI, M. (2004). Deterministic Design for Neural Network Learning : An Approach Based on Discrepancy. *In Proceedings IEEE Transactions on Neural Network*, volume 15 (13), pages 533–544.

CERVELLERA, C. et MUSELLI, M. (2006). *Advances in Imaging and Electron Physics*, volume 140, chapitre Deterministic learning and an application in optimal control, pages 61–118. CA :Academic Press.

CHANG, C.-C. et LIN, C.-J. (2001). *LIBSVM : a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

CHAPEL, L. et DEFFUANT, G. (2007). SVM viability controller active learning : application to bike control. *In IEEE Approximate Dynamic Programming and Reinforcement Learning. Hawaï (USA)*.

CHAPELLE, O. (2005). Active learning for parzen window classifier. *In AISTATS'05 : Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*.

CHENG, J. et WANG, K. (2007). Active learning for image retrieval with Co-SVM. *Pattern Recognition*, 40(1):330–334.

COHN, D., LADNER, R. et WAIBEL, A. (1994). Improving generalization with active learning. *Machine Learning*, 15:201–221.

COHN, D. A., GHAHRAMANI, Z. et JORDAN, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

CORNUÉJOLS, A. (2002). Une nouvelle méthode d'apprentissage : les SVMs. Séparateurs a vaste marge. *In Bulletin de l'AFIA - Association Francaise d'Intelligence Artificielle*, numéro 51.

DASGUPTA, S. (2006). Coarse sample complexity bounds for active learning. *In* WEISS, Y., SCHOLKOPF, B. et PLATT, J., éditeurs : *Neural Information Processing Systems (NIPS)*, volume 18, pages 235–242. MIT Press, Cambridge.

DASGUPTA, S., KALAI, A. T. et MONTELEONI, C. (2005). Analysis of perceptron-based active learning. *In COLT'O5 : Proceedings of the Conference On Learning Theory*, pages 249–263.

DE RAINVILLE, F.-M., GAGNÉ, C., TEYTAUD, O. et LAURENDEAU, D. (2009). Optimizing low-discrepancy sequences with an evolutionary algorithm. *In GECCO'11 : Proccedings of the 11th Annual Genetic and Evolutionary Computation Conference, Montreal (Canada)*.

DEHEUVELS, P., PECCATI, G. et YOR, M. (2006). On quadratic functionals of the brownian sheet and related processes. *Stochastic Processes and their Applications*, 116:493–538.

DELARA, M., DOYEN, L., GUILBAUD, T. et ROCHET, M.-J. (2007). Is a management framework based on spawning-stock biomass indicators sustainable ? A viability approach. *ICES Journal of Marine Science*, 64(4):761–767.

DEVROYE, L., GYORFI, L. et LUGOSI, G. (1997). *A Probabilistic Theory of Pattern Recognition*. Springer.

DEVROYE, L. et ROSENBLATT, M. (1982). Bounds for the uniform deviation of empirical measures. *Journal of Multivariate Analysis*, 12:72–79.

DVORETZKY, A., KIEFER, J. et WOLFOWITZ, J. (1956). Asymptotic minimax character of the sample distribution function and of a classical multinomial estimator. *Annals of Mathematical Statistics*, 33:642–669.

EDELSBRUNNER, H. (1987). *Algorithms in Combinatorial Geometry*, chapitre 2. Springer, New-York (USA).

FAURE, H. (1982). Discrépance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, 41:337–351.

FAURE, H. et LEMIEUX, C. (2008). Generalized Halton Sequences in 2008 : A comparative study.

FELDMAN, V. (2008). Statistical query learning (1993 ; kearns). The Encyclopedia of Algorithms. Springer-Verlag.

FELDMAN, V. (2009). A complete characterization of statistical query learning with applications to evolvability. *In Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 375–384, Washington, DC, USA. IEEE Computer Society.

FEUILLARD, V. (2007). *Analyse d´ une base de données pour la calibration d'un code de calcul*. Thèse de doctorat, Université Pierre et Marie Curie, et CEA (Commissariat à l'Energie Atomique et aux Energies Alternatives).

FISHER, R. A. (1951). *The design of experiments*. Oliver and Boyd.

FONTENEAU, R., MURPHY, S., WEHENKEL, L. et ERNST, D. (2010a). A cautious approach to generalization in reinforcement learning. *In ICAART'10 : Proceedings of the International Conference on Agents and Artificial Intelligence, Valencia (Espagne)*.

FONTENEAU, R., MURPHY, S., WEHENKEL, L. et ERNST, D. (2010b). Model-free monte carlo-like policy evaluation. *In* W&CP, J., éditeur : *AISTATS'10 : Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics, Chia Laguna, Sardaigne (Italie)*, volume 9, pages 217–224.

FONTENEAU, R., SUSAN A. MURPHY, S., WEHENKEL, L. et ERNST, D. (2011). Towards min max generalization in reinforcement learning. *In* FILIPE, J., FRED, A. et SHARP, B., éditeurs : *ICAART'10 : Proceedings of the International Conference on Agents and Artificial Intelligence, Valencia (Espagne)*, volume 129 de *Communications in Computed and Information Science (CCIS)*, pages 61–77. Springer, Heidelberg.

FRANCO, J. (2008). *Planification déxpèriences numériques en phase exploratoire pour la simulation des phénoménes complexes*. Thèse de doctorat, École Nationale Supérieure des Mines de Saint-Etienne.

FRANK, A. et ASUNCION, A. (2010). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

FREUND, Y., SEUNG, H. S., SHAMIR, E. et TISHBY, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (2009a). Discrépance et dispersion : des critères optimaux en apprentissage ? *In CAp'09 : Proceedings of the Conférence d'Apprentissage, Porqueroles (France)*, pages 347–351.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (2009b). How to optimize sample in active learning : Dispersion, an optimum criterion for classification ? *In ENBIS'09 : Proceedings of the Conference of the European Network for Business and Industrial Statistics, Saint-Étienne (France)*.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (2010a). Comment répartir des points pour apprendre sans *a priori* ? *In CAp'10 : Proceedings of the Conférence d'Apprentissage, Clermont-Ferrand (France)*.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (2010b). How to generate the best samples for learning in classification ? *In AISTATS'10 : Artificial Intelligence and Statistics - Workshop on Active Learning and Experimental Design, Domus de Maria, Sardinia (Italie)*.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (2011). Dispersion effect on generalisation error in classification : experimental proof and practical algorithm. *In ICAART'11 : Proceedings of the International Conference on Agents and Artificial Intelligence, Rome (Italie)*.

GANDAR, B., LOOSLI, G. et DEFFUANT, G. (subm). Why, contrarily to regression tasks, grids are the best to explore a classification pattern ? *IEEE Transactions on Pattern Analysis and Machine Intelligence*. article soumis.

GAVALDÀ, R. (1993). On the power of equivalence queries. *In EuroCOLT'93 : Proceedings of the 1rst European Conference on Computational Learning Theory*, numéro 53 de The Institute of Mathematics and its Applications Conference, pages 193–203. Oxford University Press (1994).

GILAD-BACHRACH, R., NAVOT, A. et TISHBY, N. (2006). Query by committee made real. *In Advances in Neural Network Systems (NIPS)*, volume 18, pages 443–450.

HALL, P. et MOLCHANOV, I. (2003). Sequential methods for design-adaptive estimation of discontinuities in regression curves and surfaces. *The Annals of Statistics*, 31:921–941.

HALTON, J. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerical mathematic*, 2:84–90.

HAMMERSLEY, J. (1960). Monte carlo methods for solving multivariate problems. *Annals of the New York Academy of Sciences*, 86:844–874.

HANNEKE, S. (2007). A bound on the label complexity of agnostic active learning. *In ICML'07 : Proceedings of the 24th International Conference on Machine learning*.

HICKERNELL, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67:299–322.

HOI, S. C. H., JIN, R., ZHU, J. et LYU, M. R. (2006). Batch mode active learning and its application to medical image classification. *ICML'06 : Proceedings of the 23rd international conference on Machine learning*, pages 417–424.

IWATA, K. et ISHII, N. (2002). Discrepancy as a Quality Measure for Advoiding Classification Bias. *In Proceedings of the 2002 IEEE International Symposium on Intelligent Control. Vancouver (Canada)*.

JOHNSON, M., MOORE, L. et YLVISAKER, D. (1990). Mimimax and maximin distance designs. *Journal of Statistical Planning Inference*, 26(2):131–148.

JOURDAN, A. et ZABALZA-MEZGHANI, I. (2004). Response Surface Designs for Scenario Management and Uncertainty Quantification in Reservoir Production. *Mathematical Geology*, 36(8):965 – 985.

KAARIAINEN, M. (2006). Active learning in the non-realizable case. *In ALT'06 : Proceedings of the 17th International Conferefence of Algorithmic Learning Theory*, volume 4264 de *Lecture Notes in Artificial Intelligence*, pages 63–67.

KEARNS, M. (1993). Efficient noise-tolerance learning from statistical queries. *In Proccedings of the 25th Annual ACM Symposium on the Theory of Computing, San-Diego (USA)*, pages 392–401.

KEARNS, M., SCHAPIRE, R. E., SELLIE, L. M. et HELLERSTEIN, L. (1994). Toward efficient agnostic learning. *In Machine Learning*, pages 341–352. ACM Press.

KENNARD, R. et STONE, L. (1969). Computer aided design of experiments. *Technometrics*, 11: 137–148.

KENNEDY, M. et O'HAGAN, A. (2001). Supplementary details on bayesian calibration of computer. Rapport technique, University of Nottingham, Statistics Section.

KIEFER, J. (1961). On large deviations of the empiric d.f. of vector chance variables and a law of the iterated logarithm. *Pacific Journal Mathematic*, 11:649–660.

KOROSTELEV, A. (1999). On minimax rates of convergence in image models under sequential design. *Statistics & Probability Letters*, 43(4):369 – 375.

KUIPERS, L. et NIEDERREITER, H. (1974). *Uniform distribution of sequences*. John Wiley, New York (USA).

KULKARNI, S. R. et HAUSSLER, D. (1993). Active learning using arbitrary binary valued queries. *In Machine Learning*, pages 23–35.

LAVALLE, S. et BRANICKLY, S. (2004). On the Relationship Between Classical Grid Search and Probabilistic Roadmaps. *International Journal of Robotics Research*.

L'ECUYER, P., LÉCOT, C. et TUFFIN, B. (2008). A randomized quasi-monte carlo simulation method for markov chains. *Operations Research*, 56:958–975.

L'ECUYER, P. et LEMIEUX, C. (2005). *Recent Advances in Randomized Quasi-Monte Carlo Methods*, volume 46 de *International Series in Operations Research & Management Science*, pages 419–474. Springer New York.

LEMIEUX, C. (2004). Randomized quasi-monte carlo : a tool for improving the efficiency of simulations in finance. *In WSC '04 : Proceedings of the 36th conference on Winter simulation*, pages 1565–1573. Winter Simulation Conference.

LEWIS, D. D. et GALE, W. A. (1994). A sequential algorithm for training text classifiers. *In* CROFT, W. B. et RIJSBERGEN, C. J. W., éditeurs : *SIGIR'94 :Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12. Springer-Verlag, Heidelberg.

LI, M. et VITANYI, P. B. (1993). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin.

LI, Y.-F. et ZHOU, Z.-H. (2011). Improving semi-supervised support vector machines through unlabeled instances selection. *In AAAI'11 :Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, California (USA), August 7-11*, volume 1, pages 386–391, San Francisco, CA.

LINDEMANN, S. et LAVALLE, S. (2004). Incrementally Reducing Dispersion by Increasing Voronoi Bias in RRTs. *In IEEE International Conference on Robotics and Automation*.

LINDENBAUM, M., MARKOVITCH, S. et RUSAKOV, D. (2004). Selective Sampling for Nearest Neighbor Classifiers. *Machine Learning*, 54(2):125–152.

LIU, Y. (2004). Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, 44:1936–1941.

LU, Z., WU, X. et BONGARD, J. (2010). Adaptive informative sampling for active learning. *In SDM'10 : Proceedings of the SIAM International Conference on Data Mining, April 29 - May 1, Columbus, Ohio (USA)*, pages 894–905.

LYHYAOUI, A., MARTINEZ, M., MORA, I., VÁZQUE, M., SANCHO, J.-L. et FIGUEIRAS-VIDAL, A.-. (1999). Sample selection via clustering to construct support vector-like classifiers. *IEEE Transactions On Neural Networks*, 10(6):1474–1481.

MARY, J. (2005). *Étude de l´Apprentissage Actif, Application à la Conduite d´Expèriences*. Thèse de doctorat, Université Paris XI.

MASSART, P. (1990). The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *Annals of Probability*, 18(3):1269–1283.

MCCALLUM, A. et NIGAM, K. (1998). Employing em in pool-based active learning for text classification. *In ICML'98 : Proceedings of the 15th International Conference on Machine Learning*, pages 350–358. Morgan Kaufmann.

MITCHELL, T. (1974). An algorithm for the construction of " d-optimal " experimental designs. *Technometrics*, 16(2):203–210.

MONNEROT-DUMAINE, A. (2012). Courbe du dragon. courbe de dimension topologique 1 ayant une dimension fractale de 2. Wikipédia Image. `http://fr.wikipedia.org/wiki/Fichier:Courbe_du_Dragon.gif`.

MOROKOFF, W. et CAFLISCH, R. (1995). Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122(2):218–230.

MUNGER, D., L'ECUYER, P., BASTIN, F., CIRILLO, C. et TUFFIN, B. (2012). Estimation of the mixed logit likelihood function by randomized quasi-monte carlo. *Transportation Research Part B : Methodological*, 46(2):305–320.

MUNOS, R. (2005). Geometric variance reduction in Markov chains. Application to value function and gradient estimation. *In American Conference on Artificial Intelligence*.

MUSLEA, I., MINTON, S. et KNOBLOCK, C. A. (2002). Active + semi-supervised learning = robust multi-view learning. *In ICML'02 : Proceedings of the 9th International Conference on Machine Learning*, ICML'02, pages 435–442, San Francisco, CA (USA). Morgan Kaufmann Publishers Inc.

NEDDERMEYER, J. (2011). Nonparametric particle filtering and smoothing with quasi-monte carlo sampling. *Journal of Statistical Computation and Simulation*, 81(11):1361–1379.

NGUYEN, H. T. et SMEULDERS, A. (2004). Active learning using pre-clustering. *In ICML'04 : Proceedings of the 21st international conference on Machine learning*, page 79, New York, NY (USA). ACM.

NGUYEN, X., NGUYEN, Q. et MCKAY, R. (2007). PSO with randomized low-discrepancy sequences. *GECCO '07 : Proceedings of Genetic and Evolutionary Computation Conference*, page 173.

NIEDERREITER, H. (1972). Methods for estimating discrepancy. *Applications of number theory to numerical analysis (S. K. Zaremba ed.)*, pages 203–236.

NIEDERREITER, H. (1988). Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(2):51–70.

NIEDERREITER, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics.

NIEDERREITER, H. et SPANIER, J. (1998). *Monte Carlo and Quasi-Monte Carlo Methods*.

OKTEN, G. (2009). Generalized von neumann-kakutani transformation and random-start scrambled halton sequences. *Journal of Complexity*, 25(4):318–331.

OKTEN, G. et EASTMAN, W. (2004). Randomized quasi-monte carlo methods in pricing securities. *Journal of Economic Dynamics and Control*, 28(12):2399–2426.

OKTEN, G. et WILLYARD, M. (2008). Parameterization based on randomized quasi-monte carlo methods. *In IPDPS'08 - Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium*, Miami (USA).

OUDEYER, P.-Y. et KAPLAN, F. (2004). Intelligent adaptive curiosity : a source of self-development. *In Lund University Cognitive Studies*, pages 127–130.

OWEN, A. (2004). Multidimensional Variation for Quasi-Monte Carlo. `www-stat.stanford.edu/~owen/reports/ktfang.pdf`.

PEART, P. (1982). The dispersion of the hammersley sequence in the unit square. *Monatshefte für Mathematik*, 94:249–261. 10.1007/BF01295787.

PEART, P. et MITCHELL, R. (1992). On computing the exact value of dispersion of a sequence. *Journal of Computational and Applied Mathematics*, 42(3):309–337.

PRESS, W. et TEUKOLSKY, S. (1989). Quasi- (that is, sub-) Random Numbers. *Computers in Physics*, 3(6):76 – 79.

RIESZ, F. et Nagy, B. (1955). *Functional Analysis*. Ungar Publishing Co.

RIPLEY, B. et KELLY, F. (1977). Markov point processes. *Journal of the London Mathematical Society*, 15:188–192.

ROLET, P. et TEYTAUD, O. (2010). Complexity bounds for batch active learning in classification. *In ECML PKDD'10 : Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer-Verlag, Berlin.

ROMANOWICZ, K. B. R. (2006). The future of distributed models : model calibration and uncertainty prediction. *Reliability Engineering and System Safety*, 91:1315–1231.

ROTH, K. (1954). On irregularities of distribution. *Mathematika*, 1:73–79.

ROY, N. et MCCALLUM, A. (2001). Toward Optimal Active Learning through Sampling Estimation of Error Reduction. *In ICML'01 : Proceedings of the 18th International Conference on Machine Learning*, pages 441–448. Morgan Kaufmann, San Francisco, CA (USA).

ROYLE, J. et NYCHKA, D. (1998). An algorithm for the construction of spatial coverage designs with implementation in Splus. *Computer & Geosciences*, 24(5):479–488.

SAINT-PIERRE, P. (1994). Approximation of the viability kernel. *Applied Mathematics & Optimization*, 29(2):187–209.

SAUER, N. (1972). On the density of families of sets. *Journal of Combinatorial Theory, Serie A*, 13(1):145–147.

SCHEIN, A. et UNGAR, L. (2007). Active learning for logistic regression : an evaluation. *Machine Learning*, 68(3):235–265.

SCHLIER, C. (2004). Discrepancy behaviour in the non-asymptotic regime. *Applied Numerical Matematics*, 50(2):227–238.

SCHOHN, G. et COHN, D. (2000). Less is more : Active learning with support vector machines. *In ICML'00 : Proceedings of the 17th International Conference on Machine Learning*, volume 282, pages 839–846. Morgan Kaufmann, San Francisco, CA (USA).

SERGENT, M., Phan Tan LUU, R. et ELGUERO, J. (1997). Statistical Analysis of Solvent Scales. *Anales de Quimica*, 93(Part. 1):3–6.

SETTLES, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

SETTLES, B. et CRAVEN, M. (2008). An analysis of active learning strategies for sequence labeling tasks. *In Proceedings of Empirical Methods in Natural Language Processing*, pages 1070–1079.

SEUNG, H. S., OPPER, M. et SOMPOLINSKY, H. (1992). Query by committee. *In COLT'92 : Proceedings of the 5th annual workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY (USA). ACM.

SHAWE-TAYLOR, J. et CRISTIANINI, N. (2000). *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

SINGH, A., NOWAK, R. et ZHU, X. (2008). Unlabeled data : Now it helps, now it doesn't. *In NIPS'08 : Proceedings of Neural Information Processing Systems*.

SOBOL, I. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Journal of Computational Mathematics and Mathematical Physics*, 4:86–112.

SUGIYAMA, M. et RUBENS, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Network*, 21(9):1278–1286.

SZÖRÉNYI, B. (2009). Characterizing statistical query learning : Simplified notions and proofs. *In ALT'07 : Proceedings of the 20th international conference on Algorithmic Learning Theory*, pages 186–200.

TAN, K. S. et BOYLE, P. P. (2000). Applications of randomized low discrepancy sequences to the valuation of complex securities. *Journal of Economic Dynamics and Control*, 24:1747–1782.

TEYTAUD, O., GELLY, S. et MARY, J. (2007). Active learning in regression, with application to stochastic dynamic programming. *In Proceedings of International Conference on Informatics to Control, Automation and Robotics*.

THIEMARD, E. (2001). An Algorithm to Compute Bounds for the Star Discrepancy. *Journal of Complexity*, 17(4):850 – 880.

THRUN, S. B. et MOELLER, K. (1993). Active Exploration in Dynamic Environments. *Advances in Neural Information Processing Systemes*, page 531.

TIPPING, M. E. (2001). Sparse bayesian learning and the relevance vector machin. *Journal of Machine Learning Research*, 1:211–244.

TONG, S. et KOLLER, D. (2000). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, pages 999–1006.

TOVSTIK, T. (2007). Calculation of the discrepancy of a finite set of points in the unit n-cube. *Vestnik St. Petersburg University : Mathematics*, 40:250–252. 10.3103/S1063454107030120.

TUFFIN, B. (2004). Randomization of quasi-monte carlo methods for error estimation : Survey and normal approximation. *Monte Carlo Methods and Applications*, 10(3-4):617–628.

TÜR, G., HAKKANI-TÜR, D. et SCHAPIRE, R. (2005). Combining active and semi-supervised learning for spoken language understanding. *In Speech Communication*, volume 45(2), pages 171–186.

Van der CORPUT, J. (1935). Verteilungsfunktionen. *Akad.Wetensch. Proc. Ser. B*, 38:813–821.

VAPNIK, V. (1995). *The Nature of Statistical Learning*. Springer-Verlag.

VAPNIK, V. et CHERVONENKIS, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280.

VIDYASAGAR, M. (1997). *A Theory of Learning and Generalization*. Springer-Verlag.

WANG, X. et HICKERNELL, F. J. (2000). Randomized Halton Sequences. *Mathematical and Computer Modelling*, 32:2000.

WARNOCK, T. (1972). Computational investigations of low-discrepancy point sets. *Applications of Number Theory to Numerical Analysis,(S. K. Zaremba, ed.)*, pages 319–343.

WOOTON, R., CRANFIELD, R., SHEPPEY, G. et GOODFORD, P. (1975). Physicochemical activity relationships in practice. rational selection of benzenoid substituents. *Journal of Medical Chemical*, 18:607–661.

YAKOWITZ, S., L'ECUYER, P. et VAZQUEZ-ABAD, F. (2000). Global stochastic optimization with low-dispersion point sets. *Operations Research*, 48:939–950.

YAN, R., YANG, J. et HAUPTMANN, A. (2003). Automatically labeling video data using multi-class active learning. *In Proceedings of the 9th IEEE International Conference on Computer Vision*, pages 516–523. Press.

ZHANG, C. et CHEN, T. (2002). An active learning framework for content-based information retrieval. *IEEE Transactions on multimedia*, 4(2):260–268.

ZHOU, Z.-H. et LI, M. (2005). Tri-training : Exploiting unlabeled data using three classifiers. *IEEE Transaction on Knowledge and Data Engineering*, 17(11):1529–1541.

ZHOU, Z.-H. et LI, M. (2010). Semi-supervised learning by disagreement. volume 24, pages 415–439. Springer London. 10.1007/s10115-009-0209-z.

ZHU, X., LAFFERTY, J. et GHAHRAMANI, Z. (2003). Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. *In ICML'03 : International Conference on Machine Learning - workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65.

# INDEX BIBLIOGRAPHIQUE

Kaariainen (2006), 18, 141
Kearns *et al.* (1994), 18, 141
Kearns (1993), 18, 141
Kennard et Stone (1969), 78, 81, 141
Kennedy et O'Hagan (2001), 1, 141
Kiefer (1961), 34, 141
Korostelev (1999), 112, 142
Kuipers et Niederreiter (1974), 31, 142
Kulkarni et Haussler (1993), 16, 142
L'Ecuyer *et al.* (2008), 39, 142
L'Ecuyer et Lemieux (2005), 47, 142
LaValle et Branickly (2004), 80, 142
Lemieux (2004), 39, 142
Lewis et Gale (1994), 99, 109, 142
Li et Vitanyi (1993), 11, 142
Li et Zhou (2011), 108, 142
Lindemann et LaValle (2004), 80, 83, 142
Lindenbaum *et al.* (2004), 99, 142
Liu (2004), 109, 142
Lu *et al.* (2010), 100, 142
Lyhyaoui *et al.* (1999), 100, 142
Mary (2005), 46, 47, 142
Massart (1990), 32, 142
McCallum et Nigam (1998), 104, 109, 143
Mitchell (1974), 78, 143
Monnerot-Dumaine (2012), 51, 143
Morokoff et Caflisch (1995), 55, 143
Munger *et al.* (2012), 39, 143
Munos (2005), 135, 143
Muslea *et al.* (2002), 97, 98, 102, 105, 107, 143
Neddermeyer (2011), 39, 143
Nguyen *et al.* (2007), 39, 143
Nguyen et Smeulders (2004), 100, 108, 143
Niederreiter et Spanier (1998), 30, 44, 143
Niederreiter (1972), 31, 121, 143
Niederreiter (1988), 19, 59, 60, 94, 143
Niederreiter (1992), 31–34, 39, 41, 43, 86, 143
Okten et Eastman (2004), 39, 143
Okten et Willyard (2008), 39, 143
Okten (2009), 39, 143
Oudeyer et Kaplan (2004), 105, 107, 144
Owen (2004), 54, 144
Peart et Mitchell (1992), 62, 144
Peart (1982), 62, 144
Press et Teukolsky (1989), 55, 144
Riesz (1955), 44, 144
Ripley et Kelly (1977), 86, 144
Rolet et Teytaud (2010), 15–17, 144
Romanowicz (2006), 1, 144
Roth (1954), 34, 144
Roy et McCallum (2001), 100, 101, 103, 144
Royle et Nychka (1998), 79, 144
Saint-Pierre (1994), 2, 144
Sauer (1972), 13, 144
Schein et Ungar (2007), 100, 144
Schlier (2004), 32, 144
Schohn et Cohn (2000), 105, 144
Sergent *et al.* (1997), 81, 144
Settles et Craven (2008), 109, 145
Settles (2009), 110, 144
Seung *et al.* (1992), 104, 145
Shawe-Taylor et Cristianini (2000), 54, 99, 104, 145
Singh *et al.* (2008), 108, 145
Sobol (1967), 39, 41, 145
Sugiyama et Rubens (2008), 4, 15, 145
Szörényi (2009), 18, 145
Tür *et al.* (2005), 109, 145
Tan et Boyle (2000), 39, 145
Teytaud *et al.* (2007), 19, 47, 83, 84, 145, 149
Thiemard (2001), 32, 145
Thrun et Moeller (1993), 99, 145
Tipping (2001), 99, 145
Tong et Koller (2000), 105, 109, 145
Tovstik (2007), 32, 145
Tuffin (2004), 39, 145
Van der Corput (1935), 35, 145
Vapnik et Chervonenkis (1971), 10, 51, 145
Vapnik (1995), 5, 8, 12, 53, 145
Vidyasagar (1997), 16, 146
Wang et Hickernell (2000), 39, 146
Warnock (1972), 32, 146
Wooton *et al.* (1975), 81, 146
Yakowitz *et al.* (2000), 80, 146
Yan *et al.* (2003), 109, 146
Zhang et Chen (2002), 109, 146
Zhou et Li (2005), 108, 146
Zhou et Li (2010), 108, 146
Zhu *et al.* (2003), 108, 146

# TABLE DES FIGURES

# TABLE DES TABLEAUX

# TABLE DES ALGORITHMES