# Chapitre 5

# Introduction à l'optimisation pour l'apprentissage

*Stéphane Canu, Rémi Flamary et David Mary*

## 5.1 Quelques exemples d'optimisation en apprentissage statistique

Un des objectifs de l'apprentissage statistique est de comprendre et de trouver comment améliorer automatiquement les performances d'un système grâce à des données. C'est ce que Mitchell [2006] définit par *building computer programs that automatically improve with data*. Cette idée d'amélioration est souvent reformulée comme un problème d'optimisation et ce document en contient de nombreux exemples. L'objectif de ce chapitre est, à travers ces exemples, de montrer comment aborder les questions d'optimisation liées à l'apprentissage statistique.

Avant de commencer à détailler ces exemples, nous souhaitons clarifier le cadre de ce travail. Nous avons vu dans le chapitre 2 sur les fondamentaux de l'apprentissage statistique que son but est de trouver une fonction $f$ appelée prédicteur. Nous allons faire l'hypothèse dans le reste de ce chapitre que cette fonction est spécifiée par un vecteur $\mathbf{w}$ de $d$ paramètres réels à optimiser. Cette simplification permettra de nous concentrer sur les spécificités de l'optimisation pour l'apprentissage statistique. Notons que $d$ pourra être très grand. Par exemple, la fonction de prédiction qui a gagné le challenge Imagenet avait $d = 60$ millions de paramètres à optimiser [Krizhevsky *et al.*, 2012].

Un premier exemple d'optimisation pour l'apprentissage est le LASSO (voir le paragraphe *application à la régularisation $\ell_1$ en régression* du chapitre 4). Étant donnés une matrice d'observation $\mathbf{X}$ de taille $n \times p$ et un vecteur de réponses $Y \in \mathbb{R}^n$, le LASSO s'exprime, pour un $\lambda > 0$, comme le problème

d'optimisation suivant :

$$\min_{\mathbf{w}\in\mathbb{R}^p} \quad \|\mathbf{X}\mathbf{w} - Y\|^2 + \lambda\|\mathbf{w}\|_1, \tag{5.1}$$

où $\|\mathbf{w}\|_1$ est la norme 1 (ou $\ell_1$) du vecteur $\mathbf{w}$. Ce problème peut être vu comme un cas particulier de *critère empirique pénalisé*, principe d'apprentissage formulé comme un problème d'optimisation introduit équation (2.35) dans le chapitre 2. Ce problème d'optimisation s'écrit, pour un $\lambda > 0$ donné

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \widehat{\mathcal{R}}(\mathbf{w}) + \lambda\,\Omega(\mathbf{w}), \tag{5.2}$$

où $\widehat{\mathcal{R}}$ est le risque empirique (définition 3.4 chapitre 2) vu comme un terme d'attache aux données et $\Omega$ un terme de pénalisation. Ce critère empirique pénalisé est aussi utilisé dans les réseaux de neurones (voir l'équation (7.8) chapitre 7).

Il est important en apprentissage, de prendre en compte les a priori liées à la nature du problème à traiter. Dans le cas de l'apprentissage pour la reconstruction d'images, les paramètres à optimiser sont des intensités et doivent nécessairement être positives. Le critère à optimiser doit alors intégrer ces contraintes de positivité et il s'écrit :

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & \widehat{\mathcal{R}}(\mathbf{w}) + \lambda\,\Omega(\mathbf{w}) \\ \text{avec} & \mathbf{0} \leqslant \mathbf{w}. \end{array} \tag{5.3}$$

Dans un autre exemple issu de l'astronomie, les paramètres à déterminer peuvent être vus comme la somme de deux termes : un vecteur parcimonieux lié à des impulsions et une composante continue dont la transformée en cosinus discrète (TCD) est elle aussi un vecteur parcimonieux [Bourguignon *et al.*, 2012]. Le problème d'optimisation associé s'écrit alors, pour deux paramètres $\lambda_l > 0$ et $\lambda_c > 0$ donnés

$$\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{X}\mathbf{w} - Y\|^2 + \lambda_l\|\mathbf{w}^l\|_1 + \lambda_c\|\mathbf{F}\mathbf{w}^c\|_1 \quad \text{avec } \mathbf{w} = \mathbf{w}^l + \mathbf{w}^c, \tag{5.4}$$

où $\mathbf{F}$ est l'opérateur linéaire associé à la TCD.

Un autre problème classique d'optimisation que l'on retrouve en apprentissage est celui lié aux SVM introduites dans le chapitre 6 et qui s'écrit, pour un $\lambda > 0$ donné

$$\begin{array}{lll} \min\limits_{\mathbf{w}\in\mathbb{R}^p, b\in\mathbb{R}, \xi\in\mathbb{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda\sum\limits_{i=1}^{n}\xi_i & \\ \text{avec} & Y_i(X_i\mathbf{w} + b) \geqslant 1 - \xi_i & i = 1,n \\ \text{et} & 0 \leqslant \xi_i & i = 1,n. \end{array} \tag{5.5}$$

Le problème d'approximation de faible rang est aussi formulé comme un problème d'approximation (voir équation 11.5 du chapitre 11)

$$\begin{array}{ll} \min\limits_{(\mathbf{W})\in\mathbb{R}^{n\times p}} & \|\mathbf{X} - \mathbf{W}\|_F^2 \\ \text{avec} & \text{rang}(\mathbf{W}) = k \end{array} \tag{5.6}$$

## 5.2 Le cadre général de l'optimisation

Les six problèmes d'optimisation présentés au paragraphe précédent, de (5.1) à (5.6) ont tous la forme générale suivante :

$$\mathcal{P} = \begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & J(\mathbf{w}) \\ \text{avec} & h_j(\mathbf{w}) = 0 \quad \forall\, j = 1,\ldots,\ell \\ \text{et} & g_i(\mathbf{w}) \leq 0 \quad \forall\, i = 1,\ldots,q. \end{cases} \tag{5.7}$$

La fonction $J$ est le coût ou la fonction objectif du problème. Les fonctions $h_j$ et $g_i$ permettent de définir les contraintes d'égalité et d'inégalité. On appelle *domaine admissible* ou *ensemble réalisable* du problème $\mathcal{P}$ l'ensemble des valeurs de la variable à optimiser $\mathbf{w}$ vérifiant les contraintes, soit :

$$\big\{\mathbf{w} \in \mathbb{R}^d \;\big|\; h_j(x) = 0, \forall j = 1,\ldots,\ell \text{ et } g_i(x) \leq 0, \forall i = 1,\ldots,q\big\}.$$

Le LASSO tel qu'il est introduit par l'équation (5.1) et le problème (5.2) sont dit sans contrainte puisque $\ell = q = 0$. Pour le problème (5.3) on a $\ell = 0$ et $q = n$ avec les fonctions $g_i(\mathbf{w}) = -w_i$. Le problème (5.4) s'écrit avec $\ell = p$ et $q = 0$ puisque $h_j(\mathbf{w},\mathbf{w}^l,\mathbf{w}^c) = w_j - w_j^l - w_j^c$. Pour les SVM (5.5) on a $\ell = 0$ et $q = 2n$. Enfin, pour le problème (5.6) on a $\ell = 1$ et $q = 0$ avec $h_1(\mathbf{W}) = \text{rang}(\mathbf{W}) - k$.

Mise à part cette dernière, l'ensemble des autres contraintes introduites ici sont équivalentes à imposer l'inclusion de la solution dans un ensemble convexe. Les fonctions coût des problèmes (5.1) (5.4), (5.5) et (5.6) sont convexes et celles des problèmes (5.2) et (5.3) le sont aussi le plus souvent. Cette notion de convexité joue un rôle important en optimisation et nous allons maintenant l'examiner de plus près.

### 5.2.1 Convexité

Un ensemble $\mathcal{C}$ est dit convexe si, pour tout couple $(\mathbf{w},Y) \in \mathcal{C}^2$, $\mathbf{z} \in \mathcal{C}$ pour chaque élément $\mathbf{z} = \alpha\mathbf{w} + (1-\alpha)Y$, avec $0 \leqslant \alpha \leqslant 1$. Les polyèdres sont un exemple important d'ensemble convexe. Ils sont définis comme l'ensemble des points vérifiant un nombre fini d'inégalités de la forme $\mathbf{Xw} \leqslant \mathbf{b}$ pour une matrice $\mathbf{X}$ et un vecteur $\mathbf{b}$. Les fonctions peuvent elles aussi être convexes.

**Définition 5.1** *Une fonction $J$ est dite convexe si son graphe se situe en dessous de ses cordes, c'est-à-dire si* $\forall\ \mathbf{w},Y \in \mathbb{R}^d$,

$$J(\alpha\mathbf{w} + (1-\alpha)Y) \leqslant \alpha J(\mathbf{w}) + (1-\alpha)J(Y), \text{ avec } 0 \leqslant \alpha \leqslant 1. \tag{5.8}$$

*Une fonction est dite strictement convexe lorsque les inégalités ci-dessus sont strictes.*

La figure 5.1 montre des exemples d'ensembles et de fonctions convexes et non convexes. Il existe une relation forte entre fonction convexe et ensemble convexe. En effet, si une fonction $J$ est convexe, alors l'ensemble $\{\mathbf{w} \in$

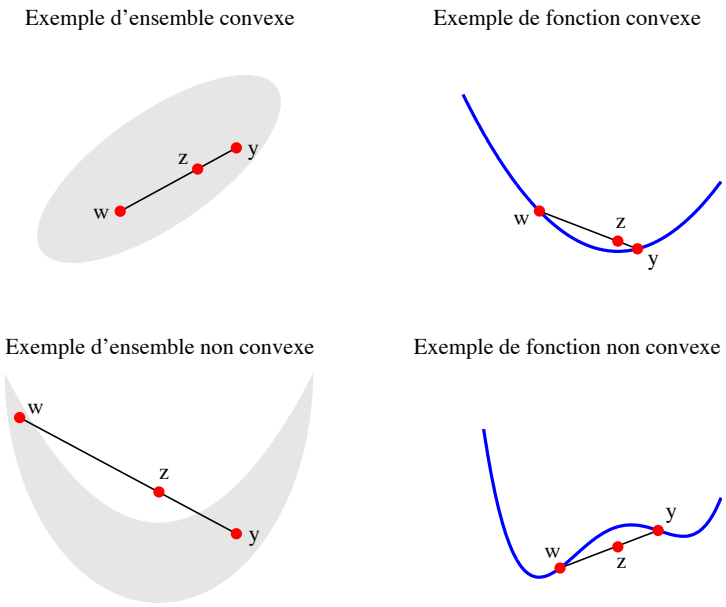


Figure 5.1 : *Illustration de la convexité d'ensembles (gauche) et de fonctions (droite).*

$\mathbb{R}^d \mid J(\mathbf{w}) \leqslant 0\}$ l'est aussi. On peut aussi représenter un ensemble convexe par une famille d'inégalités définies à partir de fonctions convexes.

Un problème d'optimisation est dit convexe lorsque sa fonction coût et son domaine réalisable sont convexes. C'est le cas en particulier quand les contraintes d'inégalités sont associées à des fonctions convexes et les contraintes d'égalités définies par des fonctions $h_j$ affines. Les problèmes (5.1), (5.4) et (5.5) sont convexes. La convexité des problèmes (5.2) et (5.3) va dépendre du choix des termes $\widehat{\mathcal{R}}$ et $\Omega$, la contrainte de positivité étant convexe. Le problème (5.6) n'est pas convexe car la contrainte de rang ne l'est pas.

De manière générale, il peut s'avérer difficile de montrer qu'un problème d'optimisation est convexe. Il existe néanmoins des règles qui permettent de conclure dans certains cas particuliers. Par exemple, la somme pondérée de fonctions convexes est convexe si les poids sont positifs. La composition d'une fonction convexe positive avec une fonction non décroissante est convexe ainsi que sa translation (pour plus de détails voir par exemple le chapitre 3 du livre de Boyd et Vandenberghe [2004]).

L'importance de la notion de convexité en optimisation est liée à la nature des questions à aborder pour traiter un problème concret. En effet, la recherche de $\mathbf{w}^\star$, la solution optimale du problème $\mathcal{P}$ (c'est-à-dire le point de coût minimal vérifiant les contraintes) pose différentes questions [Hiriart-Urruty, 2002] :

- existence et unicité de la solution,
- conditions nécessaires et suffisante d'optimalité (caractérisation de $\mathbf{w}^\star$),
- calcul de $\mathbf{w}^\star$ (les aspects algorithmiques et informatiques),
- analyse et reformulation du problème.

Quand un problème d'optimisation est convexe, l'existence et l'unicité de la solution sont généralement garantis. De plus, il existe des méthodes efficaces et relativement génériques permettant de résoudre les problèmes d'optimisation convexes. Enfin, tous les problèmes d'optimisation convexe (ou non) ne sont pas équivalents et il est possible de définir une hiérarchie parmi certains d'entre eux. Ainsi, parmi les problèmes les plus classiques on trouve les programme linéaires (LP) et les programmes quadratiques (QP) définis par :

$$\text{(LP)} \quad \begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & \mathbf{c}^\top\mathbf{w} \\ \text{avec} & \mathbf{A}\mathbf{w} \leqslant \mathbf{b} \end{cases} \qquad \text{(QP)} \quad \begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & \frac{1}{2}\mathbf{w}^\top\mathbf{G}\mathbf{w} + \mathbf{c}^\top\mathbf{w} \\ \text{avec} & \mathbf{A}\mathbf{w} \leqslant \mathbf{b} \end{cases}$$

Un programme quadratique est convexe lorsque la matrice $\mathbf{G}$ est définie positive. Un programme linéaire est convexe et est un cas particulier de QP avec $\mathbf{G} = \mathbf{0}$. Dans le cas convexe, ces problèmes sont relativement génériques et il existe des technologies bien documentées permettant de les résoudre. Il est aussi possible de s'assurer de la convexité d'un problème dès sa formulation en utilisant les règles de la programmation convexe disciplinée [*disciplined convex programming, Grant* et al*., 2006]*. Lorsqu'un problème est exprimé dans ce cadre, il peut être résolu avec la boite à outil CVX [Grant et Boyd, 2014].

### 5.2.2 Différentiabilité

Avec la convexité, la différentiabilité est une autre notion permettant de distinguer les différents types de problèmes d'optimisation. En effet, la caractérisation de la solution peut être obtenue à l'aide de la règle de Fermat qui implique le gradient lorsque $J$ est différentiable ou sinon la notion plus générale de sous différentielle.

Supposons que $J$ soit différentiable dans le sens où toutes ses dérivées partielles $\frac{\partial J}{\partial x_i}$ existent. Dans ce cas, le gradient peut être définit de la manière suivante :

**Définition 5.2 (Gradient)** *Le gradient $\boldsymbol{\nabla} J(\mathbf{w})$ d'une fonction $J$ au point $\mathbf{w}$ est le vecteur dont les composantes sont les dérivées partielles de $J$.*

**Exemple 5.1** (Moindres carrées) Soit $\mathbf{X}$ une matrice $p \times n$ et $Y$ un vecteur de réponses de talle $n$. Le gradient de la fonction coût des moindres carrés $J_1(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - Y\|^2$ est

$$\boldsymbol{\nabla} J_1(\mathbf{w}) = \mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y).$$

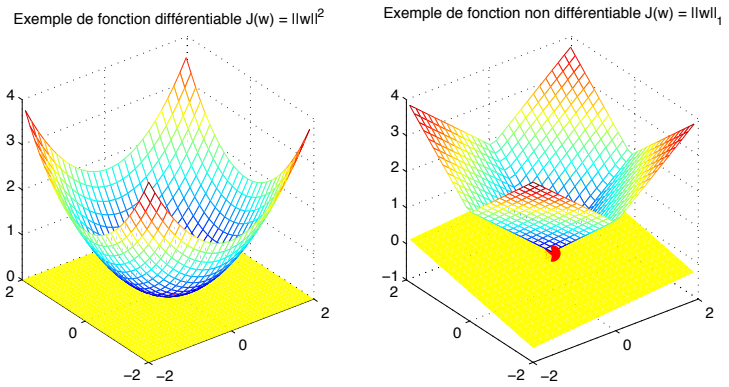


Figure 5.2 : *Exemple de sous gradient (en jaune) de fonctions différentiable (à gauche) et non-différentiable (à droite) en deux points marqués en rouge.*

**Théorème 5.1** *Si $J$ est convexe et différentiable, alors*

$$J(\mathbf{w}+\mathbf{h}) \geqslant J(\mathbf{w}) + \boldsymbol{\nabla} J(\mathbf{w})^\top \mathbf{h}, \qquad \forall\, \mathbf{h} \in \mathbb{R}^d.$$

Cette inégalité fait que le gradient peut être utilisé pour définir une borne inférieure à la fonction coût au point $\mathbf{w}$ et une direction de descente. Le gradient peut aussi être utilisé dans ce cas pour caractériser la solution globale d'un problème d'optimisation sans contraintes à travers des conditions du premier ordre (règle de Fermat) :

$$\mathbf{w}^\star = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \quad J(\mathbf{w}) \qquad \Leftrightarrow \qquad \boldsymbol{\nabla} J(\mathbf{w}^\star) = 0.$$

Un autre avantage du gradient est l'existence de règles générales facilitant son calcul (comme la règle d'additivité et la règle de chainage).

Quand $J$ n'est pas différentiable le gradient n'existe pas. Dans ce cas, la notion de sous différentielle doit être utilisée. La sous différentielle est définie à partir des sous gradients.

**Définition 5.3 (Sous gradients)** *Un vecteur $\mathbf{d} \in \mathbb{R}^d$ est un sous gradient de la fonction $J$ au point $\mathbf{w}$ si*

$$J(\mathbf{w}+\mathbf{h}) \geqslant J(\mathbf{w}) + \mathbf{d}^\top \mathbf{h}, \qquad \forall \mathbf{h} \in \mathbb{R}^d.$$

En d'autres termes, un sous gradient est un vecteur qui défini un hyperplan toujours inférieur à la fonction pour tout $\mathbf{h}$ (voir figure 5.2).

**Définition 5.4 (Sous différentielle)** *La sous différentielle $\partial J(\mathbf{w})$ de la fonction $J$ au point $\mathbf{w}$ est l'ensemble (éventuellement vide) de tous ses sous gradients en ce point*

$$\partial J(\mathbf{w}) = \left\{ \mathbf{g} \in \mathbb{R}^d \mid J(\mathbf{w}+\mathbf{d}) \geqslant J(\mathbf{w}) + \mathbf{g}^\top \mathbf{d}, \quad \forall \mathbf{d} \in \mathbb{R}^d \right\}.$$

**Exemple 5.2** Supposons $d = 1$, nous avons :

$$J_2(w) = |w| \qquad \partial J_2(0) = \{g \in \mathbb{R} \mid -1 \leqslant g \leqslant 1\},$$
$$J_3(w) = \max(0, 1 - w) \qquad \partial J_3(1) = \{g \in \mathbb{R} \mid -1 \leqslant g \leqslant 0\}.$$

La convexité de $J$ implique l'existence d'au moins un hyperplan support à chaque point de $\mathbb{R}^d$, c'est-à-dire $\partial J(\mathbf{w}) \neq \emptyset$. De plus, si $J$ est différentiable, $\boldsymbol{\nabla} J(\mathbf{w})$ est l'unique sous gradient de $J$ en $\mathbf{w}$ soit

$$\partial J(\mathbf{w}) = \{\boldsymbol{\nabla} J(\mathbf{w})\}.$$

Et donc la sous différentielle caractérise la solution globale du problème convexe car dans ce cas on a la règle de Fermat suivante :

$$\mathbf{w}^\star = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}} J(\mathbf{w}) \quad \Leftrightarrow \quad \mathbf{0} \in \partial J(\mathbf{w}^\star). \tag{5.9}$$

Quand $J$ est non différentiable et non convexe, la sous différentielle peut ne plus exister. Sa généralisation non convexe est connue comme la sous différentielle au sens de Clarke [Clarke, 1990]. Quand $J$ est différentiable et non convexe, la règle de Fermat n'est plus qu'une condition nécessaire caractérisant un minimum local.

### 5.2.3 Différents types de problèmes d'optimisation

L'analyse d'un problème d'optimisation et le choix d'un algorithme permettant de le résoudre, dépend de sa nature convexe ou non, de sa différentiabilité et de la présence de contraintes. Nous allons maintenant aborder ces trois aspects en relation avec le domaine de l'apprentissage statistiques et ses spécificités. Nous allons commencer par le cas de l'optimisation différentiable sans contraintes. Nous nous intéresserons ensuite au cas de l'optimisation différentiable avec contraintes dans le cas convexe où la règle de Fermat se généralise par les conditions de Karush-Kuhn-Tucker (KKT). Nous finirons par traiter le cas de l'optimisation non différentiable convexe et non convexe à travers certaines situations spécifiques rencontrées en apprentissage statistique.

**Exemple 5.3** Des exemples de ces situations peuvent être obtenus en considérant différents termes de pénalisation $\Omega$ dans (5.2). Souvent cette pénalité est séparable, c'est-à-dire qu'elle s'exprime comme la somme de fonctions d'une seule variable de la manière suivante :

$$\Omega(\mathbf{w}) = \sum_{i=1}^{d} \omega(w_i). \tag{5.10}$$

La figure 5.3 présente des exemples de ce type de fonctions en précisant leur caractère convexe et différentiable. Il est remarquable que, même dans le cas non différentiable, ces fonctions sont très régulières presque partout comme le montrent leurs graphes représentés à droite de la figure 5.3.

| Pénalité | Convexe | Diff. |
|---|---|---|
| $\omega(w) = w^2$<br>$\Omega(\mathbf{w}) = \|\mathbf{w}\|^2$ | ✓ | ✓ |
| $\omega(w) = 1 - \exp^{-w^2}$ | ✗ | ✓ |
| $\omega(w) = \|w\|$<br>$\Omega(\mathbf{w}) = \|\mathbf{w}\|_1$ | ✓ | ✗ |
| $\omega(w) = \sqrt{\|w\|}$<br>$\Omega(\mathbf{w}) = \|\mathbf{w}\|_{1/2}$ | ✗ | ✗ |

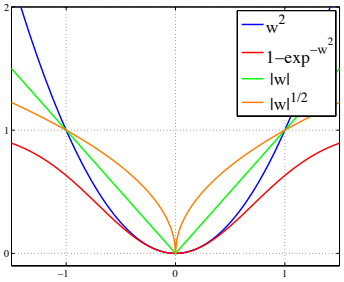


Figure 5.3 : *Exemples de différents types de pénalités utilisées en apprentissage statistique. [Pour plus d'exemples voir Antoniadis* et al.*, 2011, et les références incluses].*

## 5.3 Optimisation différentiable, convexe et sans contraintes

### 5.3.1 Théorie de l'optimisation convexe sans contraintes

Considérons maintenant le problème d'optimisation sans contraintes suivant :

$$\min_{\mathbf{w}\in\mathbb{R}^d} J(\mathbf{w}), \tag{5.11}$$

avec $J$ convexe. Nous savons que la règle de Fermat, équation (5.9), permet de caractériser les minima de cette fonction comme les zéros de sa sous différentielle ou de son gradient lorsque $J$ est différentiable.

Dans le cas où $J$ est différentiable, les méthodes de type « descente de gradient » permettent de construire des algorithmes convergeant vers la solution du problème (5.11). On peut les définir de la manière suivante.

**Définition 5.5 (Méthode de descente de gradient)** *Un algorithme de descente de gradient est un programme permettant de calculer la suite suivante :*

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)}, \tag{5.12}$$

*où* $\mathbf{d}^{(k)} \in \mathbb{R}^d$ *est une direction de descente vérifiant* $\boldsymbol{\nabla} J(\mathbf{w}^{(k)})^\top \mathbf{d}^{(k)} < 0$ *et* $\rho^{(k)} \in \mathbb{R}^+$ *un pas associé. Un choix naturel de direction de descente est, lorsqu'elle existe, la direction opposée au gradient* $\mathbf{d}^{(k)} = -\boldsymbol{\nabla} J(\mathbf{w}^{(k)})$*. Pour un choix judicieux de* $\rho^{(k)}$ *et* $\mathbf{d}^{(k)}$ *cette suite converge vers* $\mathbf{w}^\star$ *solution du problème* (5.11)*. Les conditions de convergence sont discutées en détail dans les livres de Bertsekas [1999] et Nocedal et Wright [2006]*

Le choix du pas $\rho^{(k)}$ à chaque itération peut être vu comme un problème d'optimisation en une dimension (*line search*) dans la direction $\mathbf{d}^{(k)}$. La méthode d'Armijo [Nocedal et Wright, 2006, Algorithm 3.1] est une méthode typique permettant de gérer la suite des pas. Cette méthode consiste à initialiser

$\rho$ puis à l'augmenter ou le diminuer selon que des conditions de convergence sont vérifiées ou non (conditions de Wolf) de manière à pouvoir prouver la convergence de la méthode [Nocedal et Wright, 2006, chapitre 3].

Il est intéressant de noter que la méthode du gradient peut aussi être vue comme une approche itérative minimisant à chaque itération une approximation locale du problème d'optimisation. Pour illustrer ce fait, nous proposons la définition suivante.

**Définition 5.6** *Une fonction $J$ est gradient Lispchitz si il existe une constante $L_J$ telle que*

$$\|\boldsymbol{\nabla} J(\mathbf{w}+\mathbf{d}) - \boldsymbol{\nabla} J(\mathbf{w})\| \leqslant L_J \|\mathbf{d}\|, \qquad \forall \mathbf{d} \in \mathbb{R}^d, \forall \mathbf{w} \in \mathbb{R}^d. \tag{5.13}$$

*La constante $L_J$ est appelée la constante de Lipschitz de $\boldsymbol{\nabla} J$.*

**Exemple 5.4** Le coût des moindres carrés $J_1(\mathbf{w})$ est gradient Lispchitz avec une constante $L_J = \|\mathbf{X}^\top \mathbf{X}\|$. En effet, $\boldsymbol{\nabla} J_1(\mathbf{w}+\mathbf{d}) = \mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y) + \mathbf{X}^\top \mathbf{X}\mathbf{d}$, de sorte que

$$\boldsymbol{\nabla} J_1(\mathbf{w}+\mathbf{d}) - \boldsymbol{\nabla} J_1(\mathbf{w}) = \mathbf{X}^\top \mathbf{X}\mathbf{d},$$

et

$$\|\boldsymbol{\nabla} J_1(\mathbf{w}+\mathbf{d}) - \boldsymbol{\nabla} J_1(\mathbf{w})\| \leqslant \|\mathbf{X}^\top \mathbf{X}\| \|\mathbf{d}\| = L_J \|\mathbf{d}\|.$$

Si $J$ est gradient Lispchitz, nous avons l'approximation suivante de $J$ autour de $\mathbf{w}$ (encore appelée *descent Lemma*) :

$$J(\mathbf{w}+\mathbf{d}) \leqslant J(\mathbf{w}) + \boldsymbol{\nabla} J(\mathbf{w})^\top \mathbf{d} + \frac{L_J}{2}\|\mathbf{d}\|^2, \qquad \forall \mathbf{d} \in \mathbb{R}^d, \forall \mathbf{w} \in \mathbb{R}^d. \tag{5.14}$$

Pour une preuve et plus de détails, voir [Bertsekas, 1999, Prop. A.24].

Pour trouver une direction $\mathbf{d}^{(k)}$ qui minimise l'approximation quadratique ci-dessus, on calcule le gradient du second membre par rapport à $\mathbf{d}^{(k)}$ qui s'écrit $\boldsymbol{\nabla} J(\mathbf{w}^{(k)}) + L_J \mathbf{d}^{(k)}$. En annulant le gradient, on trouve $\mathbf{d}^{(k)} = -\frac{1}{L_J}\boldsymbol{\nabla} J(\mathbf{w}^{(k)})$. Cette procédure permet de montrer que le choix d'un pas de descente $\rho^{(k)} \leqslant \frac{1}{L_J}$ garantit la décroissance stricte du coût et donc la convergence.

Supposons maintenant que $J$ soit deux fois différentiable. Il est alors possible de définir son Hessien.

**Définition 5.7 (Hessien)** *Le Hessien $\boldsymbol{\nabla}^2 J(\mathbf{w})$ d'une fonction $J$ au point $\mathbf{w}$ est la matrice de taille $n \times n$ de composantes $\nabla^2_{ij} J(\mathbf{w}) = \frac{\partial^2 J}{\partial w_i \partial w_j}(\mathbf{w})$ les dérivées partielles du second ordre de $J$.*

**Exemple 5.5** Le Hessien de $J_1$ le coût des moindre carrés de l'exemple (5.1) est

$$\boldsymbol{\nabla}^2 J_1(x) = \mathbf{X}^\top \mathbf{X}.$$

Notons que $J$ est convexe si et seulement si $\forall \mathbf{w} \in \mathbb{R}^d$, $\boldsymbol{\nabla}^2 J(\mathbf{w})$ est une matrice définie posive. A partir du gradient et du Hessien (en supposant qu'ils existent) il est possible d'utiliser l'approximation locale du second ordre suivante pour minimiser $J$ :

$$J(\mathbf{w}+\mathbf{d}) = J(\mathbf{w}) + \mathbf{d}^\top \boldsymbol{\nabla} J(\mathbf{w}) + \tfrac{1}{2}\mathbf{d}^\top \boldsymbol{\nabla}^2 J(\mathbf{w})\mathbf{d} + o(\|\mathbf{d}\|^2). \tag{5.15}$$

C'est ce que l'on appelle la méthode de Newton.

**Définition 5.8 (la méthode de Newton)** *La méthode de Newton est une méthode itérative qui consiste à minimiser, à chaque itération, l'approximation quadratique locale (5.15) de $J$ autour de $\mathbf{w}^{(k)}$. La direction résultante est alors $\mathbf{d}^{(k)} = -(\boldsymbol{\nabla}^2 J)^{-1}(\mathbf{w}^{(k)})\boldsymbol{\nabla} J(\mathbf{w}^{(k)})$. La méthode de Newton est de la forme (5.12) avec $\rho^{(k)}$ égal à un. Il est souvent possible de l'accélérer en jouant sur le pas.*

**Exemple 5.6** Les itérations des méthodes de descende de gradient et de Newton pour le problème de minimisation des moindres carrés sont :

$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \rho^{(k)}\mathbf{X}^\top(\mathbf{X}\mathbf{w}^{(k)} - Y)$, Gradient

$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\mathbf{w}^{(k)} - Y)$. Newton

Dans ce cas particulier, la méthode de Newton converge en une itération.

Une approximation du second ordre est nettement meilleure qu'une approximation de type Lispschitz, ce qui implique une convergence plus rapide des méthodes de Newton. Cependant, leur mise en œuvre requiert à chaque itération la résolution d'un système linéaire de taille $d$ ce qui peut s'avérer difficilement gérable lorsque $d$ est grand (par exemple supérieur à 10 000). Soulignons aussi qu'il existe de nombreuses autres méthodes d'optimisation de ce type (et notamment la famille des méthode type quasi Newton) que nous n'allons pas détailler car elles sont moins utilisées en apprentissage statistique.

Notons enfin que, lorsque le nombre $n$ d'exemples disponibles pour l'apprentissage est très grand (de l'ordre du million) le calcul du gradient peut lui aussi s'avérer fastidieux. Dans ce cas, il peut s'avérer plus efficace de n'utiliser qu'une approximation stochastique du gradient comme direction de descente [pour plus de détails voir par exemple Bottou, 2010]. C'est le principe de ce type de méthode qui est généralement utilisé pour optimiser la fonction $J$ non convexe des réseaux de neurones à couches (voir chapitre 7). Le choix concret d'une méthode dépend alors du compromis entre coût de calcul d'une direction de descente et les gains espérés en terme de diminution du coût qui définira *in fine* sa vitesse de convergence [voir par exemple Bubeck, 2015, et les références incluses].

### 5.3.2 Exemple : la régression logistique

La régression logistique binaire est une méthode populaire de discrimination bi classe et une exemple classique d'optimisation sans contrainte [pour plus de

détails voir par exemple Hastie *et al.*, 2005]. Pour $\mathbf{X}$ une matrice d'exemples donnés de taille $n \times p$ et un vecteur d'étiquettes[1] $Y \in \{0,1\}^n$, la régression logistique consiste à résoudre le problème d'optimisation suivant

$$\min_{\mathbf{w}\in\mathbb{R}^p} \; J_\ell(\mathbf{w}) = \sum_{i=1}^{n} \left(-Y_i(\mathbf{X}\mathbf{w})_i + \log\bigl(1+\exp(\mathbf{X}\mathbf{w})_i\bigr)\right), \tag{5.16}$$

avec $(\mathbf{X}\mathbf{w})_i$ la $i^{\text{ème}}$ composante du vecteur $\mathbf{X}\mathbf{w}$. Le coût $J_\ell$ est convexe et différentiable. Il peut être interprété comme l'opposé de la log vraisemblance d'un échantillon de Bernoulli $\{Y_i, i = 1,n\}$ de paramètres $p_i(\mathbf{w}) = \exp((\mathbf{X}\mathbf{w})_i)/(1 + \exp((\mathbf{X}\mathbf{w})_i))$. Le problème (5.16) est une reformulation du principe de maximum de vraisemblance.

La fonction coût $J_\ell(\mathbf{w})$ étant la composition de fonctions deux fois différentiables, son gradient et sa hessienne existent tous les deux et sont :

$$\begin{array}{l} \boldsymbol{\nabla} J_\ell(x) = \mathbf{X}^\top(\mathbf{p} - Y) \\ \boldsymbol{\nabla}^2 J_\ell(x) = \mathbf{X}^\top\mathbf{P}\mathbf{X}, \end{array} \tag{5.17}$$

avec $\mathbf{p} \in \mathbb{R}^p$ le vecteur de composantes $p_i$ et $\mathbf{P}$ une matrice diagonale de terme général $\mathbf{P}_{ii} = p_i(1 - p_i), i = 1,n$. A partir du vecteur gradient et de la matrice hessienne, les itérations de Newton construisent la suite suivante :

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - (\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{p} - Y).$$

La valeur du vecteur $\mathbf{w}^{(k+1)}$ peut aussi s'exprimer à partir de $\mathbf{w}^{(k)}$ comme la solution d'un problème de moindres carrés pondérés [voir Hastie *et al.*, 2005]. Dans ce cas,

$$\begin{array}{rcl} \mathbf{w}^{(k)} - (\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{p} - Y) & = & (\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\bigl((\mathbf{X}^\top\mathbf{P}\mathbf{X})\mathbf{w}^{(k)} - \mathbf{X}^\top(\mathbf{p} - Y)\bigr) \\ & = & (\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{P}\mathbf{z}, \end{array}$$

avec $\mathbf{z} = \mathbf{X}\mathbf{w}^{(k)} + \mathbf{P}^{-1}(Y - \mathbf{p})$. L'algorithme 5.1, met en œuvre cette solution.

## 5.4 Optimisation différentiable, convexe et avec contraintes

### 5.4.1 Théorie de l'optimisation avec contraintes, convexe et différentiable

Dans le cas général de l'optimisation sous contraintes définie par un problème $(\mathcal{P})$ convexe et différentiable, les conditions nécessaires et suffisantes d'optimalité sont données par les conditions de Karush, Kuhn and Tucker (KKT).

[1] Attention, les calculs avec des étiquettes $Y \in \{-1,1\}^n$ sont analogues mais différents.

**Algorithm 5.1** Méthode de Newton pour la régression logistique

**Entrées** : $\mathbf{X}$,$Y$ données d'apprentissage.
**Sortie** : $\mathbf{w}$ paramètres du modèle
$\mathbf{w} \leftarrow 0$ initialisation des paramètres
**while** on n'a pas convergé **Faire**
$\mathbf{p} \leftarrow \dfrac{\exp^{\mathbf{Xw}}}{1+\exp^{\mathbf{Xw}}}$ a division terme à terme
$P_{ii} \leftarrow p_i(1-p_i), \quad i=1,n$ équation (5.17)
$\mathbf{z} \leftarrow \mathbf{Xw} + \mathbf{P}^{-1}(Y-\mathbf{p})$
$\mathbf{w} \leftarrow \left(\mathbf{X}^\top \mathbf{P}\mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{P}\,\mathbf{z}$
**Fin while**

**Définition 5.9 (Conditions de Karush, Kuhn and Tucker (KKT))** *Les vecteurs* ($\mathbf{w} \in \mathbb{R}^d$, $\boldsymbol{\mu} \in \mathbb{R}^p$,$\boldsymbol{\nu} \in \mathbb{R}^q$) *vérifient les conditions KKT du problème* $\mathcal{P}$ *si :*

$$
\begin{array}{lll}
\textit{stationnarité} & \boldsymbol{\nabla} J(\mathbf{w}) + \displaystyle\sum_{j=1}^{\ell} \mu_j \boldsymbol{\nabla} h_j(\mathbf{w}) + \sum_{i=1}^{q} \nu_i \boldsymbol{\nabla} g_i(\mathbf{w}) = \mathbf{0} & \\
\textit{admissibilité primale} & h_j(\mathbf{w}) = 0 & j = 1,\ldots,\ell \\
 & g_i(\mathbf{w}) \leq 0 & i = 1,\ldots,q \\
\textit{admissibilité duale} & \nu_i \geq 0 & i = 1,\ldots,q \\
\textit{complémentarité} & \nu_i g_i(\mathbf{w}) = 0 & i = 1,\ldots,q.
\end{array}
$$

*$\mu_j$ et $\nu_i$ sont appelés les multiplicateurs de Lagrange du problème $\mathcal{P}$.*

Les variables $\mathbf{w}$ sont aussi appelées les variables primales et $\boldsymbol{\mu}$,$\boldsymbol{\nu}$ les variables duales. L'intuition associée à la condition de stationnarité est en rapport avec la règle de Fermat puisqu'il s'agit d'annuler une combinaison de gradients. La condition de complémentarité traduit le fait que, s'il n'est pas possible de trouver une solution réalisable qui annule juste le gradient $\boldsymbol{\nabla} J$, il est alors nécessaire de trouver un point à la frontière du domaine admissible. Cette situation est illustrée par l'exemple suivant :

**Exemple 5.7 (Les conditions KKT dans un exemple simple)** Nous considérons le problème d'optimisation en une dimension :

$$\min_{w \in \mathbb{R}} \quad \tfrac{1}{2}(w+1)^2 \quad \text{avec} \quad w \geqslant 0,$$

admet les conditions KKT suivantes :

$$
\begin{array}{ll}
\text{stationnarité} & (w+1) - \nu = 0 \\
\text{admissibilité primale} & -w \leqslant 0 \\
\text{admissibilité duale} & \nu \geq 0 \\
\text{complémentarité} & \nu w = 0.
\end{array}
$$

La condition de complémentarité impose que $w$ ou $\nu$ soit nul. S'il s'agit de $\nu$, alors la condition de stationnarité donne $w = -1$, valeur non admissible. Donc la solution est $w = 0$ et $\nu = 1$. On dit alors que la condition d'admissibilité primale (la contrainte) est saturée ou active.

**Exemple 5.8 (Moindres carrés positifs)** Il s'agit d'un cas particulier du problème (5.3) avec le coût des moindres carrés et sans pénalisation. Le problème s'écrit alors :

$$\begin{cases} \min\limits_{\mathbf{w}\in\mathbb{R}^n} & \frac{1}{2}\|\mathbf{X}\mathbf{w} - Y\|^2 \\ \text{avec} & 0 \leqslant w_i, \qquad i = 1,\ldots,p \end{cases}$$

Les conditions KKT de ce problème sont :

| | |
|---|---|
| stationnarité | $\mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y) - \nu = \mathbf{0}$ |
| admissibilité primale | $-\mathbf{w} \leqslant \mathbf{0}$ |
| admissibilité duale | $\nu \geq \mathbf{0}$ |
| complémentarité | $\mathrm{diag}(\nu)\mathbf{w} = \mathbf{0}$. |

**Théorème 5.2** *[Theorem 12.1 dans Nocedal et Wright, 2006] Un vecteur* $\mathbf{w}^\star$ *est solution du problème d'optimisation* $\mathcal{P}$ *supposé convexe et différentiable, c'est-à-dire son minimum global s'il existe, des multiplicateurs de Lagrange* $\{\mu_j^\star\}_{j=1,\ldots,\ell}$, $\{\nu_i^\star\}_{i=1,\ldots,q}$ *tels que le triplet* $\left(x^\star, \{\mu_j^\star\}_{j=1,\ldots,\ell}, \{\nu_i^\star\}_{i=1,\ldots,q}\right)$ *vérifient les conditions KKT.*

Dans le cas non convexe, ces conditions caractérisent uniquement un point stationnaire. Afin de calculer les conditions de stationnarité, il est pratique d'introduire le lagrangien du problème $\mathcal{P}$.

**Définition 5.10** *Le Lagrangien* $\mathcal{L}$ *du problème* $\mathcal{P}$ *est la fonction suivante :*

$$\mathcal{L}(\mathbf{w},\boldsymbol{\mu},\boldsymbol{\nu}) = J(\mathbf{w}) + \sum_{j=1}^{\ell} \mu_j h_j(\mathbf{w}) + \sum_{i=1}^{q} \nu_i g_i(\mathbf{w}). \tag{5.18}$$

L'utilisation du lagrangien facilite le calcul des conditions de stationnarité car elle sont données par $\boldsymbol{\nabla}\mathcal{L}(\mathbf{w}^\star,\boldsymbol{\mu},\boldsymbol{\nu}) = \mathbf{0}$. De plus, la solution du problème d'optimisation est un point selle du Lagrangien et est donnée par

$$\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w},\boldsymbol{\mu},\boldsymbol{\nu}).$$

**Exemple 5.9 (Lagrangien de l'exemple 5.8)**

$$\mathcal{L}(\mathbf{w},\boldsymbol{\nu}) = \tfrac{1}{2}\|\mathbf{X}\mathbf{w} - Y\|^2 - \boldsymbol{\nu}^\top\mathbf{w}.$$

**Exemple 5.10 (L'exemple 5.2 vu du point de vue du lagrangien)** Considérons le problème d'optimisation sous contrainte suivant, formulé avec les notations de l'exemple (5.2) et pour un $k > 0$ donné :

$$\left\{ \begin{array}{ll} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & \widehat{\mathcal{R}}(\mathbf{w}) \\ \text{avec} & \Omega(\mathbf{w}) \leqslant k, \end{array} \right. \tag{5.19}$$

avec $\widehat{\mathcal{R}}$ et $\Omega$ deux fonctions convexes. Le lagrangien de ce problème est

$$\mathcal{L}(\mathbf{w},\nu) = \widehat{\mathcal{R}}(\mathbf{w}) + \nu\big(\Omega(\mathbf{w}) - k\big),$$

de sorte que, pour un $k$ donné, il existe un multiplicateur de Lagrange $\nu$ solution du problème (vérifiant les KKT) qui fasse que les formulations (5.2) et (5.19) sont équivalentes, dans le sens où elles admettent la même solution. Ce même raisonnement peut être formulé à partir de problème suivant :

$$\left\{ \begin{array}{ll} \min\limits_{\mathbf{w}\in\mathbb{R}^d} & \Omega(\mathbf{w}) \\ \text{avec} & \widehat{\mathcal{R}}(\mathbf{w}) \leqslant \epsilon, \end{array} \right. \tag{5.20}$$

qui est aussi équivalent aux deux autres formulations. Soulignons que cette équivalence est liée à la nature convexe du problème.

La notion de dualité est une notion importante et utile en optimisation que l'on peut définir à partir du lagrangien. Ainsi la fonction objective duale de Lagrange $Q$ est définie à partir du Lagrangien

$$\begin{aligned} Q(\boldsymbol{\mu},\boldsymbol{\nu}) &= \inf_{\mathbf{w}\in\mathbb{R}^n} \ \mathcal{L}(\mathbf{w},\boldsymbol{\mu},\boldsymbol{\nu}) \\ &= \inf_{\mathbf{w}\in\mathbb{R}^n} \ J(\mathbf{w}) + \sum_{j=1}^{\ell} \mu_j h_j(\mathbf{w}) + \sum_{i=1}^{q} \nu_i g_i(\mathbf{w}) \end{aligned}$$

et le problème d'optimisation dual associé est :

**Définition 5.11 (Problème d'optimisation dual)** *Le problème d'optimisation dual $\mathcal{P}$ est*

$$\mathcal{D} = \left\{ \begin{array}{ll} \max\limits_{\boldsymbol{\mu}\in\mathbb{R}^\ell,\boldsymbol{\nu}\in\mathbb{R}^q} & Q(\boldsymbol{\mu},\boldsymbol{\nu}) \\ \textit{avec} & \nu_j \geqslant 0, \qquad j = 1,\ldots,q \end{array} \right.$$

**Exemple 5.11 (Calcul du problème dual de l'exemple 5.8)**

$$Q(\boldsymbol{\nu}) = \inf_{\mathbf{w}\in\mathbb{R}^d} \ \tfrac{1}{2}\|\mathbf{X}\mathbf{w} - Y\|^2 - \boldsymbol{\nu}^\top \mathbf{w}$$

Le gradient du lagrangien par rapport à $\mathbf{w}$ est $\mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y) - \boldsymbol{\nu}$ ce qui nous donne $\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}(\boldsymbol{\nu} + \mathbf{X}^\top Y)$. En injectant ce résultat dans le lagrangien on obtient

$$Q(\boldsymbol{\nu}) = -\tfrac{1}{2}(\mathbf{X}^\top Y + \boldsymbol{\nu})(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top Y + \boldsymbol{\nu})$$

et le problème dual est

$$\begin{cases} \max\limits_{\boldsymbol{\nu}\in\mathbb{R}^q} & -\frac{1}{2}\boldsymbol{\nu}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\boldsymbol{\nu} - Y^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\boldsymbol{\nu} \\ \text{avec} & \nu_j \geqslant 0, \qquad j = 1,\ldots,q, \end{cases}$$

qui est aussi un QP avec des contraintes de positivité. Soulignons que grâce à la condition de stationnarité de l'exemple (5.8) $\boldsymbol{\nu} = \mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y)$, le problème dual peut être reformulé à travers les variables primales comme :

$$\begin{cases} \max\limits_{\mathbf{w}\in\mathbb{R}^d} & -\frac{1}{2}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} \\ \text{avec} & \mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y) \geqslant 0. \end{cases}$$

Enfin, le résultat suivant permet de s'assurer, sous certaines conditions, qu'il est équivalent de résoudre les problèmes primal ou dual.

**Théorème 5.3 (Saut de dualité)** *(voir les théorèmes 12.12, 12.13 et 12.14 dans Nocedal & Wright pp 346) Si J,g et h sont convexes et continûment différentiable, sous certaines conditions dites de qualification de contraintes, le coût de la solution duale à l'optimum est le même que le coût de la solution du problème primal.*

Quel que soit le vecteur $\mathbf{w}$ admissible, on a $Q(\boldsymbol{\mu},\boldsymbol{\nu}) \leqslant J(\mathbf{w})$. Cette différence entre les coûts primal et dual est appelé le saut de dualité. Cette quantité est toujours positive et si elle est égale à zéro, c'est que nous sommes en présence des solutions des problèmes primal et dual.

### 5.4.2 Exemple : la boule minimum englobante ou SVDD

Nous allons maintenant illustrer les différents points présentés ci-dessus par le problème de recherche de la boule minimum englobante. Ce problème a été introduit dans le domaine de l'apprentissage par Tax et Duin [2004] sous le nom de *support vector data description* (SVDD) comme un estimateur d'une ligne de niveau d'une densité de probabilité.

Étant donné un ensemble de $n$ observations $\{X_i \in \mathbb{R}^p, i = 1,\ldots,n\}$ ce problème consiste à trouver la boule $p$ dimensionnelle de centre $\mathbf{c}$ de rayon minimum $R$ contenant tous les points $X_i$. Il peut être exprimé par le problème d'optimisation suivant :

$$\begin{cases} \min\limits_{R\in\mathbb{R},\mathbf{c}\in\mathbb{R}^p} & R^2 \\ \text{avec} & \|X_i - \mathbf{c}\|^2 \leqslant R^2, \quad i = 1,\ldots,n. \end{cases} \tag{5.21}$$

En posant $\rho = \|\mathbf{c}\|^2 - R^2$, son lagrangien est :

$$\mathcal{L}(\mathbf{c},\rho,\boldsymbol{\nu}) = \|\mathbf{c}\|^2 - \rho + \sum_{i=1}^{n} \nu_i\big(\|X_i\|^2 - 2\mathbf{c}^\top X_i + \rho\big),$$

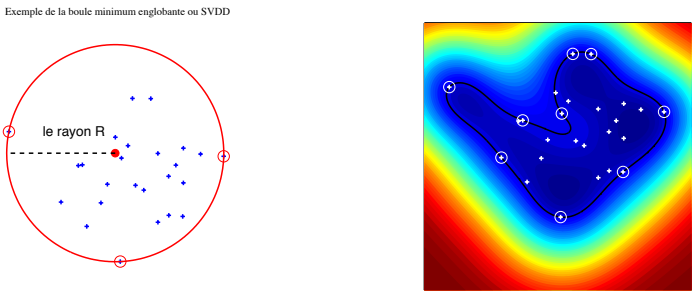


Figure 5.4 : *Exemple en deux dimensions de boule englobante de rayon minimal (en rouge à gauche) et de cette même boule kernelisée avec un noyau gaussien (en noir à droite). Les points entourés sont des observations support.*

avec les conditions KKT :

$$\begin{array}{lll}
\text{stationnarité} & 2\mathbf{c} - 2\sum_{i=1}^{n} \nu_i X_i = 0 & \\
 & -1 + \sum_{i=1}^{n} \nu_i = 0 & \\
\text{admissibilité primale} & 2\mathbf{c}^\top X_i \geqslant \rho + \|X_i\|^2 & i = 1, \ldots, n \\
\text{admissibilité duale} & \nu_i \geqslant 0 & i = 1, \ldots, n \\
\text{complémentarité} & \nu_i\big(2\mathbf{c}^\top X_i - \rho - \|X_i\|^2\big) = 0 & i = 1, \ldots, n.
\end{array}$$

Les conditions de complémentarité font que l'on peut distinguer deux ensembles de points : les points support se situant exactement sur le cercle (pour lesquels $\|X_i - \mathbf{c}\|^2 = R^2$) et les points à l'intérieur du cercle pour lesquels $\nu_i = 0$. Les conditions de stationnarité nous apportent le théorème de représentation :

$$\mathbf{c} = \sum_{i=1}^{n} \nu_i X_i. \tag{5.22}$$

qui permet de représenter le centre du cercle $\mathbf{c}$ comme une combinaison linéaire des observations support.

Après quelques calculs fastidieux, on obtient la forme duale du problème (5.21), avec $\mathbf{G} = \mathbf{X}\mathbf{X}^\top$ la matrice de Gram de terme général $G_{ij} = X_i^\top X_j$,

$$\left\{ \begin{array}{lll}
\min\limits_{\boldsymbol{\nu} \in \mathbb{R}^n} & \boldsymbol{\nu}^\top \mathbf{G} \boldsymbol{\nu} - \boldsymbol{\nu}^\top \mathrm{diag}(\mathbf{G}) & \\
\text{avec} & \boldsymbol{e}^\top \boldsymbol{\nu} = 1 & \\
\text{et} & 0 \leqslant \nu_i & i = 1, \ldots, n.
\end{array} \right. \tag{5.23}$$

Ce problème est un programme quadratique (QP) et donc le primal l'est aussi.

Les avantages et les inconvénients des formulations primale et duale sont résumés dans la table 5.1. La figure 5.4 (à gauche) illustre un exemple de SVDD en deux dimensions.

Table 5.1 : *Avantages et les inconvénients des formulations primales et duales des SVDD*

| Primal (5.21) | Dual (5.23) |
|---|---|
| $p+1$ inconnues | $n$ inconnues |
| $n$ contraintes | $p$ contraintes de boite |
| peut être reformulé comme un QP | exige la construction de $\mathbf{G}$ la matrice des influences deux à deux |
| préférable lorsque $p < n$ | à utiliser lorsque $p > n$ |

Ce modèle SVDD (5.21) est limité car il n'admet pas d'erreur et ne permet que la construction de cercles. Nous allons maintenant voir comment étendre le modèle initial pour traiter ces deux cas de figure.

Il est possible de relaxer le modèle (5.21) pour prendre en compte les erreurs éventuelles en introduisant des variables d'écart $\xi_i, i = 1, \ldots, n$ associées à chaque observation et définies par

$$\forall i = 1,n \quad \left\{ \begin{array}{rll} \text{pas d'erreur} & \|X_i - \mathbf{c}\|^2 \leqslant R^2 \Rightarrow & \xi_i = 0 \\ \text{erreur} & \|X_i - \mathbf{c}\|^2 > R^2 \Rightarrow & \xi_i = \|X_i - \mathbf{c}\|^2 - R^2. \end{array} \right.$$

En utilisant ces variables d'écart, le problème initial des SVDD (5.21) est généralisé par, pour un paramètre $C > 0$ donné

$$\left\{ \begin{array}{lll} \min\limits_{R \in \mathbb{R}, \mathbf{c} \in \mathbb{R}^p, \boldsymbol{\xi} \in \mathbb{R}^n} & R^2 + C \sum\limits_{i=1}^{n} \xi_i & \\ \text{avec} & \|X_i - \mathbf{c}\|^2 \leqslant R^2 + \xi_i & i = 1, \ldots, n \\ & 0 \leqslant \xi_i & i = 1, \ldots, n. \end{array} \right. \tag{5.24}$$

Ce problème se réduit au SVDD sans bruit (5.21) lorsque $C \to \infty$. Ces SVDD généralisées sont associées au le problème dual

$$\left\{ \begin{array}{lll} \min\limits_{\boldsymbol{\nu} \in \mathbb{R}^n} & \boldsymbol{\nu}^\top \mathbf{G} \boldsymbol{\nu} - \boldsymbol{\nu}^\top \mathrm{diag}(\mathbf{G}) & \\ \text{avec} & \sum_{i=1}^{n} \nu_i = 1 & \\ \text{et} & 0 \leqslant \nu_i \leqslant C & i = 1, \ldots, n. \end{array} \right. \tag{5.25}$$

Soulignons que l'introduction des variables d'écart modifie peu le problème dual qui reste un QP avec contraintes de boites. Elle ajoute seulement une contrainte de boite supplémentaire sur les variables duales.

Un moyen efficace pour introduire des non linéarités dans le modèle et aller plus loin que le simple cercle, consiste à utiliser des fonctions noyaux positifs comme caractéristiques. Le résultat est illustré figure 5.4 (à droite). Dans ce cas, grâce aux noyaux, le modèle circulaire a été déformé pour mieux s'ajuster aux données. Dans ce cadre, un noyau est une fonction de deux variables définie positive. Les exemples les plus courants de noyaux sont les noyaux gaussien $k_g$

avec $h > 0$ leur largeur de bande et $k_r$ le noyau positif d'ordre $r$

$$k_g(\mathbf{x},\mathbf{x}') = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{h}\right), \qquad k_r(\mathbf{x},\mathbf{x}') = (1+\mathbf{x}^\top\mathbf{x}')^r. \tag{5.26}$$

A chaque noyau on peut associer une norme dans un espace *ad hoc* $\|\cdot\|_{\mathcal{H}}$ et l'utiliser pour définir la version dite *kernelisée* des SVDD [pour plus de détail sur les machines à noyau voir Smola et Schölkopf, 1998] :

$$\left\{\begin{array}{lll} \min\limits_{R\in\mathbb{R},\mathbf{c}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^n} & R^2 + C\sum\limits_{i=1}^{n}\xi_i & \\ \text{avec} & \|k(\mathbf{x},X_i)-\mathbf{c}(\mathbf{x})\|^2_{\mathcal{H}} \leqslant R^2+\xi_i & i=1,\ldots,n \\ & 0\leqslant\xi_i & i=1,\ldots,n. \end{array}\right. \tag{5.27}$$

Cette définition (5.27) des SVDD *kernelisée* est très similaire à la définition des SVDD initiales (5.25). La seule différence est que la notion de boule englobante s'applique au sens de la norme $\|\cdot\|_{\mathcal{H}}$ associée avec le noyau. De plus, le problème dual lui reste presque quasiment inchangé puisque la seule différence entre le dual de la version *kernelisée* et (5.25) tient dans le calcul de la matrice de Gram qui devient avec les noyaux $G_{ij} = k(X_i,X_j)$. Dans ce cas, le problème primal peut s'exprimer dans une très grande dimension (voir infinie avec le noyau gaussien) et donc difficile voire impossible à résoudre alors que le problème dual reste un QP de dimension $n$. Connaissant les variables duales $\nu_i, i = 1,\ldots,n$ et $\rho$, le théorème de représentation (5.22) donne $\mathbf{c}(X) = \sum_{i=1}^n \nu_i k(X,X_i)$ de sorte que l'appartenance à la boule englobante de rayon minimum soit définie par le signe de

$$\begin{aligned} \|k(X,.)-\mathbf{c}(.)\|^2_{\mathcal{H}} - R^2 &= \|k(X,.)\|^2_{\mathcal{H}} - 2\langle k(X,.),\mathbf{c}(.)\rangle_{\mathcal{H}} + \|\mathbf{c}(.)\|^2_{\mathcal{H}} - R^2 \\ &= -2\mathbf{c}(X) + k(X,X) + \rho \\ &= -2\sum_{i=1}^{n}\nu_i k(X,X_i) + k(X,X) + \rho. \end{aligned}$$

## 5.5 Optimisation sans contraintes non différentiable

### 5.5.1 La méthode proximale pour l'optimisation non différentiable

La non convexité est un vaste domaine dont nous n'allons explorer qu'une petite partie en rapport avec l'apprentissage statistique. Nous allons nous concentrer sur le cas où la fonction à minimiser est de la forme $J(\mathbf{w}) = \widehat{\mathcal{R}}(\mathbf{w}) + \lambda\Omega(\mathbf{w})$ comme nous l'avons introduit (5.2). Une telle fonction est dite composite. Nous allons nous restreindre au cas où $\widehat{\mathcal{R}}$ est différentiable et $\Omega$ une pénalité non-différentiable. Ce type de problème apparait relativement fréquemment dans

le domaine de l'apprentissage statistique notamment dans les applications de traitement du signal et des images dans le cas où l'on cherche à promouvoir la parcimonie, c'est-à-dire quand on recherche une solution $\mathbf{w}$ qui a peu de composantes non nulles [Bach *et al.*, 2011].

Il existe différentes manières d'aborder la non différentiabilité. On peut utiliser la notion de sous différentielle [Clarke, 1990], proposer des relaxations différentiables [Antoniadis *et al.*, 2011] ou utiliser une méthode proximale que nous allons maintenant détailler [Combettes et Pesquet, 2011]. L'approche proximale propose de minimiser itérativement une succession de majoration de la fonction objectif. Supposons que la fonction $\widehat{\mathcal{R}}(\mathbf{w})$ soit Lipschitz gradient, alors à l'itération $k$, grâce au *descent lemma* (5.14), nous avons

$$J(\mathbf{w}) \leqslant \widehat{\mathcal{R}}(\mathbf{w}^{(k)}) + \boldsymbol{\nabla}\widehat{\mathcal{R}}(\mathbf{w}^{(k)})^\top(\mathbf{w}-\mathbf{w}^{(k)}) + \frac{1}{2\rho}\|\mathbf{w}-\mathbf{w}^{(k)}\|^2 + \lambda\Omega(\mathbf{w}), \quad (5.28)$$

lorsque $\rho \leqslant \frac{1}{L_L}$ avec $L_L$ la constante de Lipschitz de la fonction $\widehat{\mathcal{R}}(\mathbf{w})$. Cette majoration peut être minimisée lorsque la pénalité $\Omega(\mathbf{w})$ a une bonne forme, par exemple lorsqu'elle est séparable au sens de (5.10). Dans ce cas, elle s'écrit :

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad \tfrac{1}{2}\|\mathbf{w}-\mathbf{u}\|^2 + \lambda\rho\Omega(\mathbf{w}),$$

avec $\mathbf{u} = \mathbf{w}^{(k)} - \rho\boldsymbol{\nabla}\widehat{\mathcal{R}}(\mathbf{w}^{(k)})$. L'expression ci-dessus est associée à l'opérateur proximal dont la définition est :

**Définition 5.12 (Opérateur proximal)** *L'opérateur proximal de la fonction* $\Omega$ *est :*

$$\begin{aligned} prox_{\Omega} : \mathbb{R}^d &\longrightarrow \mathbb{R}^d \\ \mathbf{w} &\longmapsto prox_{\Omega}(\mathbf{w}) = \underset{\mathbf{u}\in\mathbb{R}^d}{argmin}\ \Omega(\mathbf{u}) + \tfrac{1}{2}\|\mathbf{u}-\mathbf{w}\|^2. \end{aligned}$$

Comme nous l'avons illustré ci-dessus, l'opérateur proximal est un élément clé de la méthode proximale pour minimiser les fonctionnelles composites. Cette méthode est la suivante.

**Définition 5.13 (Algorithme de descente de gradient proximal)** *Un principe général pour résoudre le problème d'optimisation (5.2) consiste à mettre en œuvre les itérations suivantes*

$$\mathbf{w}^{k+1} = prox_{\rho^{(k)}\lambda\Omega}\big(\mathbf{w}^k - \rho^{(k)}\boldsymbol{\nabla}\widehat{\mathcal{R}}(\mathbf{w}^k)\big).$$

*C'est l'algorithme de descente de gradient proximal, aussi connu sous le nom de* Forward Backward splitting *dans la communauté de traitement du signal.*

Le pas $\rho$ peut être fixé *a priori* à partir de la connaissance de la constante de Lipschitz ou adapté à chaque itération pour accélérer la convergence. L'efficacité

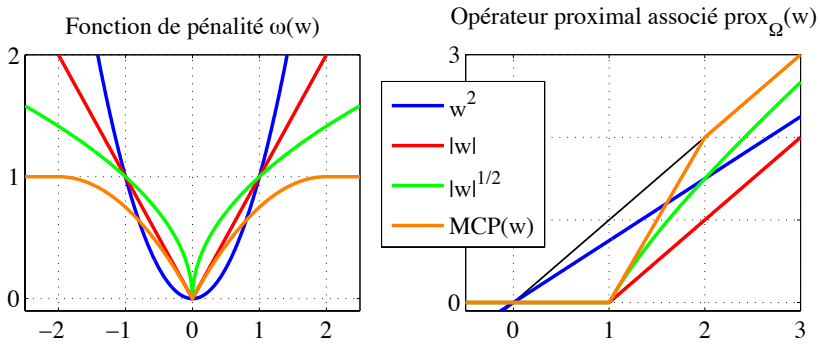


Figure 5.5 : *Illustration de différentes pénalités (à gauche) et de leur opérateur proximal (à droite).*

de la méthode en terme de vitesse de convergence dépend aussi de la nature de la pénalité $\Omega(\mathbf{w})$. En particulier, quand $\Omega(\mathbf{w})$ est séparable comme dans l'équation (5.10), l'opérateur proximal est facile à calculer et l'algorithme est efficace. L'exemple suivant présente quelques exemples d'opérateurs proximaux associés à des pénalités séparables. La figure 5.5 illustre ces pénalités et les opérateurs proximaux associés en une dimension, pour $\lambda = 1$.

**Exemple 5.12 (Opérateurs proximaux utilisés en apprentissage)**

| | | |
|---|---|---|
| $\Omega(\mathbf{w}) = 0$ | $\text{prox}_\Omega(\mathbf{w}) = \mathbf{w}$ | identité |
| $\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_2^2$ | $\text{prox}_\Omega(\mathbf{w}) = \frac{1}{1+\lambda}\mathbf{w}$ | rétrécissement |
| $\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$ | $\text{prox}_\Omega(\mathbf{w}) = \text{sign}(\mathbf{w})\max(0, \lvert\mathbf{w}\rvert - \lambda)$ | seuillage doux |
| $\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_{1/2}^{1/2}$ | [Xu *et al.*, 2012b, Equation 11] | *bridge or power family* |
| $\Omega(\mathbf{w}) = \mathbb{1}_C(\mathbf{w})$ | $\text{prox}_\Omega(\mathbf{w}) = \underset{\mathbf{u}\in C}{\text{argmin}}\ \frac{1}{2}\|\mathbf{u} - \mathbf{w}\|^2$ | seuillage dur ou projection. |

La dernière fonction de l'exemple 5.12 est une fonction indicatrice, c'est-à-dire une fonction dont la valeur est $+\infty$ pour tous les $\mathbf{w}$ situées à l'extérieur de $C$. Il est donc possible d'utiliser ce type de méthode pour résoudre des problèmes d'optimisation avec contraintes en utilisant cette fonction indicatrice et la projection comme opérateur proximal.

Si le pas $\rho$ est bien choisi, il est possible de démontrer que l'algorithme défini en 5.13 converge vers un minimum global, s'il existe (par exemple dans le cas convexe). Pratiquement, pour obtenir une meilleure convergence il est recommandé d'utiliser la règle de Barzilai-Borwein qui utilise une estimation locale de la courbure de la fonction coût [Wright *et al.*, 2009]. Comme pour de nombreuses méthodes itérative, il est aussi possible d'accélérer l'algorithme du gradient proximal [pour plus de détails voir Nesterov *et al.*, 2007; Beck et Teboulle, 2009].

Dans le cas non convexe, sous certaines conditions vérifiées par les pénalités données en exemple, l'algorithme proximal converge vers un point stationnaire

c'est à dire en général vers un minimum local [Attouch *et al.*, 2010]. Du code Matlab mettant en œuvre cette méthode sur des problèmes non convexes est disponible en ligne à `github.com/rflamary/nonconvex-optimization`.

### 5.5.2 L'exemple de l'approximation parcimonieuse

#### Les moindres carrés parcimonieux

Le Lasso introduit par l'équation (5.1) est un exemple de problème d'optimisation sans contrainte convexe non différentiable. Sa fonction coût est composite avec $\widehat{\mathcal{R}}(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - Y\|^2$ et $\Omega(\mathbf{w}) = \sum_{i=1}^{p} |w_i|$ la norme $\ell_1$ du vecteur des inconnues. L'opérateur proximal associé est la somme de termes indépendants composante par composante de la forme seuillage doux

$$\begin{aligned} \mathrm{prox}_{\Omega}(\mathbf{w}) &= \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^{p} |u_i| + \tfrac{1}{2}\|\mathbf{u} - \mathbf{w}\|^2 \\ &= \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^{p} \big(|u_i| + \tfrac{1}{2}(u_i - w_i)^2\big). \end{aligned}$$

Comme la valeur absolue n'est pas différentiable en zéro, il faut calculer sa sous différentielle qui est (voir la fonction $J_2$ de l'exemple 5.2)

$$\partial(|u_i| + \tfrac{1}{2}(u_i - w_i)^2) = \begin{cases} sign(u_i) & + u_i - w_i & & \text{si } u_i \neq 0 \\ g & + u_i - w_i & \text{avec } -1 \leqslant g \leqslant 1 & \text{si } u_i = 0, \end{cases}$$

de sorte que

$$0 \in \partial(|u_i| + \tfrac{1}{2}(u_i - w_i)^2) \quad \Leftrightarrow \quad u_i = \begin{cases} sign(u_i)(|w_i| - 1) & \text{si } |w_i| > 1 \\ 0 & \text{si } |w_i| \leqslant 1. \end{cases}$$

En utilisant ce résultat dans la définition 5.13, on obtient la méthode du gradient proximal pour le LASSO mise en œuvre par l'algorithme 5.2.

La condition d'optimalité ci-dessus explique pourquoi la pénalité $\ell_1$ promeut la parcimonie. Lorsque la fonction coût est non différentiable en zéro, la condition pour avoir une composante $u_i$ nulle n'est pas une égalité mais une inclusion, ce qui donne la possibilité de vérifier cette condition d'optimalité avec tous les éléments d'un ensemble. Cette inclusion est dans un sens plus facile à obtenir qu'une égalité car elle est vérifiée dès que $w_i$ est plus petit qu'un seuil. De plus, le fait que l'opérateur proximal produise des vecteurs parcimonieux permet souvent d'accélérer le calcul du gradient en ne calculant que ses composantes non nulles.

#### Un exemple d'optimisation non convexe : la régression logistique parcimonieuse

La pénalité de type *minimax concave penalty* (MCP) est un exemple de terme non différentiable et non convexe utilisé en apprentissage [Breheny et Huang,

**Algorithm 5.2** L'algorithme de gradient proximal pour le LASSO

**Entrées** : $\mathbf{X}$,$Y$ données d'apprentissage.
**Sortie** : $\mathbf{w}$ paramètres du modèle
$\rho \leftarrow 1/\|\mathbf{X}^\top \mathbf{X}\|$ initialisation du pas
**while** on n'a pas convergé **Faire**
  $\mathbf{w} \leftarrow \mathbf{w} - \rho \mathbf{X}^\top(\mathbf{X}\mathbf{w} - Y)$ pas de gradient (*forward*)
  $\mathbf{w} \leftarrow sign(\mathbf{w}) \max(0, |\mathbf{w}| - \rho\lambda)$ opérateur proximal (*backward*)
**Fin while**

2011]. Elle est définie, pour un couple ($\lambda \geqslant 0$,$\gamma \geqslant 1$) d'hyperparamètres, par la somme, pour chacune des variables, de la fonction suivante :

$$\Omega_{\lambda,\gamma}(t) = \begin{cases} \lambda t - \dfrac{t^2}{2\gamma} & \text{si } t \leqslant \gamma\lambda \\ \dfrac{\gamma\lambda^2}{2} & \text{sinon.} \end{cases}$$

Le paramètre $\lambda$ gère l'équilibrage entre le terme d'attache aux données et la pénalité. Le paramètre $\gamma$ contrôle la forme de la pénalité comme l'illustre la figure 5.6. Cette pénalité MCP peut être vue comme une amélioration de la pénalité $\ell_1$ utilisée par exemple dans le LASSO (5.1), car cette dernière peut provoquer un biais significatif en minorant dans certains cas les gros coefficients. La pénalité MCP bénéficie elle d'une propriété oracle de type asymptotique car il est possible de montrer qu'elle donne des estimations aussi bonnes que celles obtenues en sachant à l'avance parmi les coefficients lesquels sont nuls et lesquels ne le sont pas [Breheny et Huang, 2011].

Afin d'illustrer notre propos, nous allons étudier la pénalité MCP avec le risque empirique de type logistique $J_\ell$ tel que nous l'avons défini par l'équation (5.16). Cette pénalité bien utilisée, devrait améliorer les performances en généralisation de notre estimateur et promouvoir sa parcimonie. Le problème d'optimisation associé à la forme générale donnée par l'équation (5.2) est défini par

$$\min_{\mathbf{w}\in\mathbb{R}^p} \sum_{i=1}^{n} \Big( -Y_i(\mathbf{X}\mathbf{w})_i + \log\big(1 + \exp(\mathbf{X}\mathbf{w})_i\big) \Big) + \sum_{i=1}^{p} \Omega_{\lambda,\gamma}(|w_i|), \tag{5.29}$$

Un moyen pour traiter la non différentiabilité et la non convexité consiste à décomposer la pénalité MCP comme la différence de deux fonctions, l'une non convexe mais différentiable et l'autre convexe mais non différentiable. Le problème d'optimisation (5.29) se réécrit alors

$$\min_{\mathbf{w}\in\mathbb{R}^p} \underbrace{J_\ell(\mathbf{w}) - \lambda \sum_{j=1}^{p} h_{\lambda,\gamma}(|w_j|)}_{J_m(\mathbf{w})} + \lambda\|\mathbf{w}\|_1, \tag{5.30}$$

avec

$$h_{\lambda,\gamma}(t) = \left\{ \frac{t^2}{2\gamma\lambda} \mathbb{1}_{\{t \leq \gamma\lambda\}} + \left( t - \frac{\gamma\lambda}{2} \right) \mathbb{1}_{\{t > \gamma\lambda\}} \right\},$$

la fonction de Huber de paramètre $\gamma\lambda$ et illustrée figure 5.6.

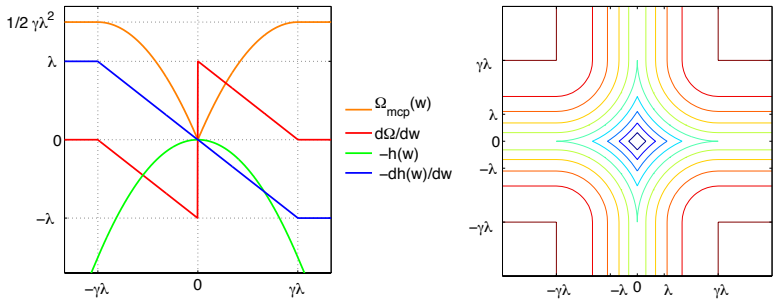


Figure 5.6 : *A gauche : la fonction $\Omega_{\lambda,\gamma}(t)$ (en orange) pour $\lambda = 1$ et $\gamma = 3$, sa dérivée (en rouge), l'inverse de la fonction coût de Huber (en vert) et sa dérivée (en bleu). A droite : des lignes de niveaux de la pénalité MCP.*

Dans ce problème d'optimisation sans contraintes, le coût à minimiser est la somme de deux termes. Le premier terme $J_m(\mathbf{w})$ est non convexe mais différentiable. Le second terme est la norme $\ell_1$ des paramètres qui est convexe mais non différentiable. Cette non différentiabilité peut être gérée en utilisant l'algorithme proximal. Le gradient de $J_m$ est (voir la courbe verte de la figure 5.6)

$$\boldsymbol{\nabla}_\beta J_m(\mathbf{w}) = \mathbf{X}^\top(\mathbf{p} - Y) - \left\{ \begin{array}{ll} sign(w_j)\lambda & \text{si } |w_j| > \lambda\gamma \\ \dfrac{w_j}{\gamma} & \text{sinon.} \end{array} \right.$$

L'opérateur proximal de la norme $\ell_1$ est

$$\text{prox}_{\ell_1}(u) \;=\; \left\{ \begin{array}{ll} 0 & \text{si } |u| \leqslant \lambda \\ sign(u)(|u| - \lambda) & \text{sinon.} \end{array} \right.$$

La procédure globale du gradient proximal est donnée par l'algorithme 5.3. Pour une valeur du pas bien choisie, par exemple $\rho \leqslant \frac{1}{\sigma_M^2}$, $\sigma_M$ étant la plus grande valeur singulière de la matrice $\mathbf{X}$, cet algorithme converge vers un minimum local du problème (5.29). En effet, de part la non convexité du problème, la convergence globale ne peut pas être garantie par l'utilisation d'un algorithme proximal.

Soulignons enfin que la décomposition (5.30) n'est pas unique et qu'il est donc possible de dériver d'autres algorithmes proximaux pour résoudre ce problème.

**Algorithm 5.3** L'algorithme proximal $\ell_1$ pour la régression logistique MCP

**Entrées** : $\mathbf{X}$,$Y$ données d'apprentissage.
**Sortie** : $\mathbf{w}$ paramètres du modèle
$\rho \leftarrow 1/\|\mathbf{X}^\top\mathbf{X}\|$ initialisation du pas
**while** on n'a pas convergé **Faire**
  $\mathbf{p} \leftarrow \frac{\exp^{\mathbf{Xw}}}{1+\exp^{\mathbf{Xw}}}$ division terme à terme
  $\mathbf{w} \leftarrow \mathbf{w} - \rho\big(\mathbf{X}^\top(\mathbf{p}-Y) - sign(\mathbf{w})\min(\lambda, \frac{|\mathbf{w}|}{\gamma})\big)$
  $\mathbf{w} \leftarrow sign(\mathbf{w})\max(0, |\mathbf{w}| - \rho\lambda)$ l'opérateur $\text{prox}_{\ell_1}$
**Fin while**

## 5.6 Conclusion

Ce chapitre ne constitue qu'une courte introduction aux principales problématiques de l'optimisation pour l'apprentissage et ne fait qu'effleurer la surface de ce domaine par ailleurs très actif. En même temps, comme l'ont souligné Parmee et Hajela [2012] *numerical optimization is now a mature technology,* dans le sens où il existe des logiciels standardisés, robustes et suffisamment rapides pour résoudre des problèmes concernant des millions de variables et pouvant être utilisé dans le domaine de l'apprentissage statistique. Aussi le développement d'une application d'optimisation devrait suivre les étapes suivantes. D'abord il faut commencer par identifier les variables, la fonction cout et les contraintes éventuelles. Ensuite, il est souvent utile de reformuler le problème pour en expliciter la nature (convexe, différentiable, LP, QP...) afin de faciliter l'utilisation de logiciels, libres comme CVX, SeDuMi, GLPK (pour les programmes linéaires), openOpt (en python), Optaplanner (en java) ou propriétaires tels que Gurobi, Cplex, Mosek ou Xpress pour ne nommer qu'eux[2].

A ce stade il nous semble utile de donner au lecteur qui souhaite approfondir le sujet une liste de références plus spécifiques. L'optimisation convexe (avec ou sans contraintes) est un domaine bien exploré et nous renvoyons à Boyd et Vandenberghe [2004] pour un exposé complet et très pédagogique au sujet, ainsi qu'aux livre de Bertsekas [1999] et Nocedal et Wright [2006]. Le livre de Bubeck [2015], disponible à partir du site web de l'auteur, discute des méthodes de gradient conditionnel et d'optimisation stochastique qui apparaissent dans les problèmes de très grande taille.

L'optimisation non différentielle et les méthodes proximales sont présentés en détail par Combettes et Pesquet [2011] et plus récemment par Parikh et Boyd [2014]. Pour une étude de ces algorithmes en relation avec le domaine de l'apprentissage et les différentes formes de parcimonies nous recommandons les travaux de Bach *et al.* [2011]. Enfin, les problèmes de convergence et la théorie des opérateurs proximaux (et monotone) sont bien détaillés par Bauschke et

[2]Pour plus de détails et comparaisons se reporter à `plato.asu.edu/bench.html`.

Combettes [2011].

Nous avons aussi la conviction que dans un futur pas si éloigné, les progrès en optimisation nous permettront d'aborder de nouveaux types de problèmes que nous n'avons pas traité ici comme les problèmes d'optimisation mixte associant variables discrètes et continues, qui apparaissent en apprentissage statistique dans les contraintes de rang ou dans les pénalités de type $\ell_0$.