

Optimization for machine learning

Introduction to optimization

Stéphane Canu

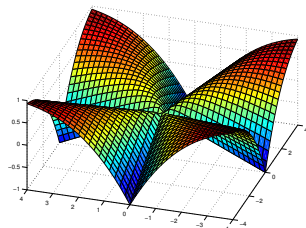
asi.insa-rouen.fr/enseignants/~scanu

S4D 2018 : Research Summer School on Statistics for Data Science

Caen, June 18, 2018

Plan

- 1 Examples of optimization for machine learning
- 2 Unconstrained convex and differentiable optimization
- 3 Constrained convex and differentiable optimization
- 4 Nondifferentiable unconstrained optimization



Examples of optimization for machine learning

LASSO, given X , Y and $\lambda > 0$

$$\min_{\mathbf{w} \in \mathbb{R}^p} \underbrace{\|\mathbf{X}\mathbf{w} - Y\|^2 + \lambda \|\mathbf{w}\|_1}_{J(\mathbf{w})} \quad (1)$$

Minimization of the penalized empirical risk , given $\lambda > 0$

$$\min_{\mathbf{w} \in \mathbb{R}^d} \underbrace{L(\mathbf{w}) + \lambda \Omega(\mathbf{w})}_{J(\mathbf{w})} \quad (2)$$

L is the empirical risk (data loss)

Ω penalization (prior)

Optimization without constraints

$$\min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) \quad \begin{array}{l} J: \mathbb{R}^d \longrightarrow \mathbb{R} \\ \mathbf{w} \longmapsto J(\mathbf{w}) \end{array}$$

with positivity constraints (**inequality** constraints)

$$\left\{ \begin{array}{l} \min_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) + \lambda \Omega(\mathbf{w}) \\ \text{with } \underbrace{\mathbf{0} \leq \mathbf{w}}_{g_1(\mathbf{w}) = -\mathbf{w}} \end{array} \right. . \quad (3)$$

for two given parameters $\lambda_l > 0$ and $\lambda_c > 0$ (**equality**) constraints

$$\left\{ \begin{array}{l} \min_{\mathbf{w}, \mathbf{w}^l, \mathbf{w}^c \in \mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{Y}\|^2 + \lambda_l \|\mathbf{w}^l\|_1 + \lambda_c \|\mathbf{F}\mathbf{w}^c\|_1 \\ \text{with } \underbrace{\mathbf{w} = \mathbf{w}^l + \mathbf{w}^c}_{h_j(\mathbf{w}) = \mathbf{w} - \mathbf{w}^l - \mathbf{w}^c} \end{array} \right. \quad (4)$$

Optimization (with constraints)

$$\mathcal{P} = \left\{ \begin{array}{l} \min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) \\ \text{with } h_j(\mathbf{w}) = 0 \quad \forall j = 1, \dots, \ell \\ \text{et } g_i(\mathbf{w}) \leq 0 \quad \forall i = 1, \dots, q \end{array} \right. \quad \begin{array}{l} h_j, g_i : \mathbb{R}^d \longrightarrow \mathbb{R} \\ \mathbf{w} \longmapsto J(\mathbf{w}) \end{array}$$

Examples of optimization in machine learning

SVM, given $\lambda > 0$

$$\left\{ \begin{array}{l} \min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) \\ \text{with } h_j(\mathbf{w}) = 0 \quad \forall j = 1, \dots, \ell \\ \text{et } g_i(\mathbf{w}) \leq 0 \quad \forall i = 1, \dots, q \end{array} \right.$$

$$\left\{ \begin{array}{l} \min_{\mathbf{w} \in \mathbb{R}^p, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \sum_{i=1}^n \xi_i \\ \text{with } Y_i(X_i \mathbf{w} + b) \geq 1 - \xi_i \quad i = 1, n \\ \text{and } 0 \leq \xi_i \quad i = 1, n. \end{array} \right. \quad (5)$$

$$\begin{aligned} g_i(\mathbf{w}) &= 1 - \xi_i - Y_i(X_i \mathbf{w} + b) & i = 1, n \\ g_i(\mathbf{w}) &= -\mathbf{w} & i = n + 1, 2n \end{aligned}$$

Low rank approximation

$$\begin{array}{l} \min_{\mathbf{W} \in \mathbb{R}^{n \times p}} \|\mathbf{X} - \mathbf{W}\|_F^2 \\ \text{with } \text{rang}(\mathbf{W}) = k \end{array} \quad (6)$$

Optimization issues

$$\mathcal{P} = \begin{cases} \mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^d} & J(\mathbf{w}) \\ \text{with} & h_j(\mathbf{w}) = 0 \quad \forall j = 1, \dots, \ell \\ \text{and} & g_i(\mathbf{w}) \leq 0 \quad \forall i = 1, \dots, q. \end{cases} \quad (7)$$

The road map: [HU02]

- **existence** uniqueness of the solution,
- **optimality conditions d'optimalité** (characterization of the solution \mathbf{w}^*),
- computation of \mathbf{w}^* (**computational issues**),
- analysis et **reformulation**.

Two fundamental properties

- convexity
- differentiability

The return of the LASSO

$$\min_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \sum_{j=1}^p |w_j|$$

CVX for matlab cvxr.com/cvx/, CVX for python cvxopt.org/

```
cvx_begin
variables w(p)
  minimize( .5*sum((yi - Xi*beta1).^2) + lambda * sum(abs(w)) )
cvx_end
```

$$\left\{ \begin{array}{l} \min_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \sum_{j=1}^p |w_j| \\ \text{with } \mathbf{0} \leq \mathbf{w} \end{array} \right.$$

```
cvx_begin
variables w(p)
  minimize( .5*sum((yi - Xi*beta1).^2) + lambda * sum(abs(w)) )
  subject to
    w >= 0;
cvx_end
```

CVX: the Swiss knife



What is the best optimization software?

CVX	commercial software (Cplex, GLPK...)	specific code
-----	--------------------------------------	---------------

the LASSO with CVX

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^P} & \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \sum_{j=1}^P |w_j| \\ \text{with} & \mathbf{0} \leq \mathbf{w} \\ \text{and} & \|\mathbf{X}\mathbf{w} - \mathbf{y}\| \geq k \end{cases}$$

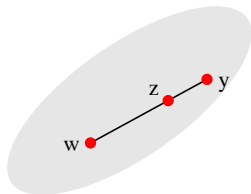
```
cvx_begin
variables w(p)
minimize( .5*sum((yi - Xi*beta1).^2) + lambda * sum(abs(w)) )
subject to
    w >= 0;
    norm(Y-X*w) >= k;
cvx_end
```

```
Error using cvxprob/newcnstr (line 192)
Disciplined convex programming error:
    Invalid constraint: {convex} >= {real constant}
...
Cannot minimize a(n) concave expression.
```

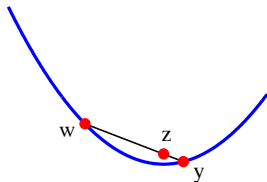
the importance of being convex

Convexity

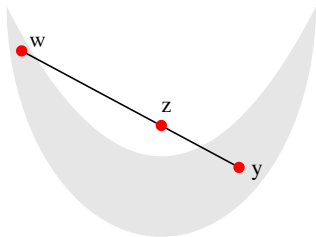
Exemple d'ensemble convexe



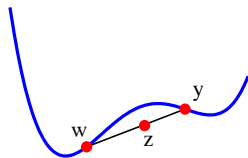
Exemple de fonction convexe



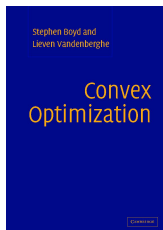
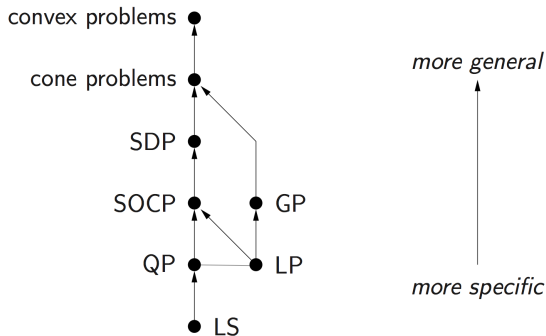
Exemple d'ensemble non convexe



Exemple de fonction non convexe



Hierarchy of Convex Optimization Problems



LS least square

LP linear programming (quantile regression)

QP quadratic programming (SVM)

GP geometric programming (Logistic regression)

SOCP second order cone programming (Group lasso)

Hierarchy of Convex Optimization Problems

linear programming (LP) quadratic programming (QP)

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

convexe (c) when \mathbf{G} is positive definite.

The CVX class

\mathcal{P} is convex if J and the g_i are convex and equality constraints are affine
 $h_j(\mathbf{w}) = \mathbf{a}_j^\top \mathbf{w} + b_j = 0$

Disciplined Convex Programming hierarchy

LP \subset (c)QP \subset (c)QCQP \subset SOCP \subset SDP \subset CVX

CVX [GB14] also supports geometric programming (GP) and mixed integer disciplined convex programming (MIDCP) .

Differentiability: the gradient

Assume partial differential $\frac{\partial J}{\partial x_i}$ exist.

Definition (Gradient)

the gradient $\nabla J(\mathbf{w})$ of a function J at point \mathbf{w} is the vector whose components are partial derivatives of J .

Example

(Least square) Let \mathbf{X} be a $p \times n$ matrix and Y a vector of size n . The gradient of the cost function $J_1(\mathbf{w}) = \frac{1}{2} \|\mathbf{X}\mathbf{w} - Y\|^2$ is

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

Fermat's rule

Theorem

If J is convex and differentiable, then

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

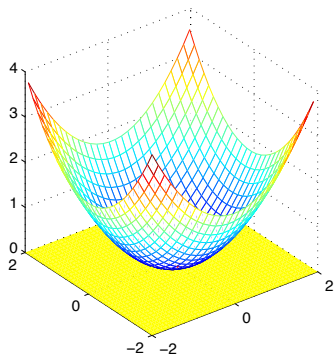
First order optimality conditions (Fermat's rule)

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) \quad \Leftrightarrow \quad \nabla J(\mathbf{w}^*) = 0.$$

Furthermore: linearity, chain and product rule make it easy to compute

the non differentiable case

Exemple de fonction différentiable $J(w) = \|w\|^2$



Exemple de fonction non différentiable $J(w) = \|w\|_1$

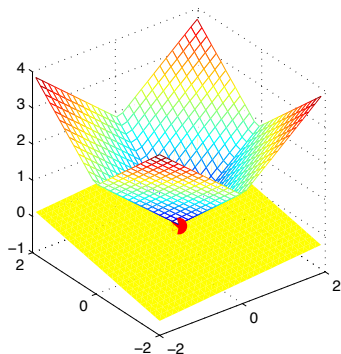


Figure : Example of sub gradient (yellow) of differentiable functions (left) et non-differentiable (right).

Definition (Sub gradient)

A vector $\mathbf{d} \in \mathbb{R}^d$ is a sub gradient of function J at point \mathbf{w} if

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

Definition (Sub differential)

The sub differential $\partial J(\mathbf{w})$ of function J at point \mathbf{w} is the set (possibly empty) of all its sub gradient at this point

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

Example

for $d = 1$ we have:

$$J_2(w) = |w|$$

$$\partial J_2(0) = \{g \in \mathbb{R} \mid -1 \leq g \leq 1\},$$

$$J_3(w) = \max(0, 1 - w)$$

$$\partial J_3(1) = \{g \in \mathbb{R} \mid -1 \leq g \leq 0\}.$$

Fermat's rule

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) \Leftrightarrow \mathbf{0} \in \partial J(\mathbf{w}^*). \quad (8)$$

Table : Fermat's rule

	differentiable	non differentiable
cvx	$\mathbf{w}^* = \arg \min_{\mathbf{w}} J(\mathbf{w}) \Leftrightarrow \nabla J(\mathbf{w}^*) = 0$	$\mathbf{w}^* = \arg \min_{\mathbf{w}} J(\mathbf{w}) \Leftrightarrow 0 \in \partial J(\mathbf{w}^*)$
non cvx	$\mathbf{w}^* = \arg \min_{\mathbf{w} \in V(\mathbf{w}_0)} J(\mathbf{w}) \Rightarrow \nabla J(\mathbf{w}^*) = 0$	$\mathbf{w}^* = \arg \min_{\mathbf{w} \in V(\mathbf{w}_0)} J(\mathbf{w}) \Rightarrow 0 \in \partial_c J(\mathbf{w}^*)$

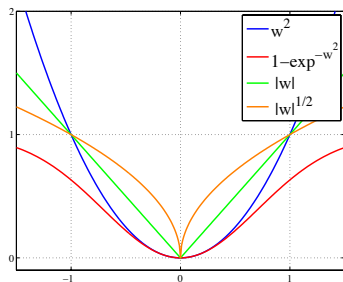
global vs. local Optimum

Example (Additive loss with penalties)

$$\min_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) + \lambda \Omega(\mathbf{w}),$$

$$\Omega(\mathbf{w}) = \sum_{i=1}^d \omega(w_i). \quad (9)$$

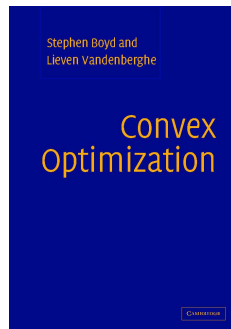
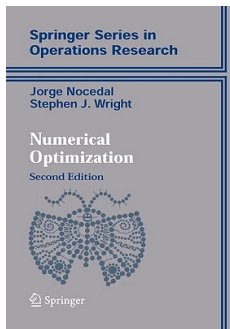
Penalty	Convex	Diff.
$\omega(w) = w^2$ $\Omega(\mathbf{w}) = \ \mathbf{w}\ ^2$	✓	✓
$\omega(w) = 1 - \exp^{-w^2}$	✗	✓
$\omega(w) = w $ $\Omega(\mathbf{w}) = \ \mathbf{w}\ _1$	✓	✗
$\omega(w) = \sqrt{ w }$ $\Omega(\mathbf{w}) = \ \mathbf{w}\ _{1/2}$	✗	✗



Non convex but some times quasi convex

Plan

- 1 Examples of optimization for machine learning
- 2 Unconstrained convex and differentiable optimization
- 3 Constrained convex and differentiable optimization
- 4 Nondifferentiable unconstrained optimization



Differentiable Optimization (without constraint)

$$\min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}), \quad (10)$$

with J differentiable and convex.

Definition (Gradient descent)

A gradient descent algorithm computes the following sequence:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \rho^{(k)} \mathbf{d}^{(k)}, \quad (11)$$

- $\mathbf{d}^{(k)} \in \mathbb{R}^d$ descent direction $\nabla J(\mathbf{w}^{(k)})^\top \mathbf{d}^{(k)} < 0$
- A natural choice $\mathbf{d}^{(k)} = -\nabla J(\mathbf{w}^{(k)})$.
- $\rho^{(k)} \in \mathbb{R}^+$ the associated stepsize
- For a good choice of $\rho^{(k)}$ and $\mathbf{d}^{(k)}$ this sequence converges towards \mathbf{w}^* the solution of the problem (10) [Ber99] et [NW06]

The choice of the step-size $\rho^{(k)}$ is a line search optimization problem (in 1 d)

Illustration 2d

$$J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top A \mathbf{w} - \mathbf{b}^\top \mathbf{w}$$

Iso cost lines : $\{ \mathbf{w} \in \mathbb{R}^2 \mid J(\mathbf{w}) = \text{Cte} \}$

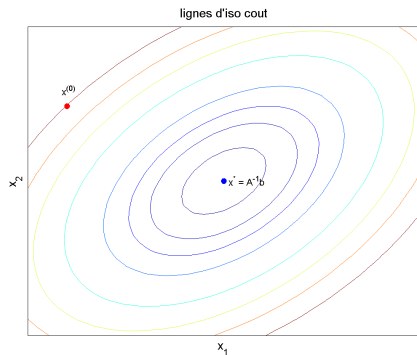
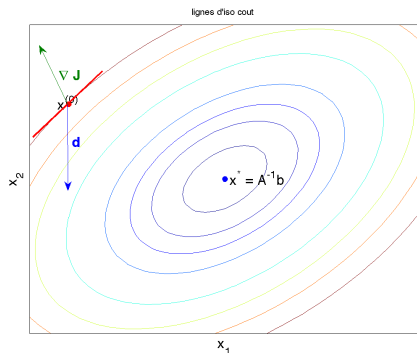


Illustration 2d

$$J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top A \mathbf{w} - \mathbf{b}^\top \mathbf{w}$$

Iso cost lines : $\{ \mathbf{w} \in \mathbb{R}^2 \mid J(\mathbf{w}) = \text{Cte} \}$



a descent direction \mathbf{d} have to verify: $\mathbf{d}^\top \underbrace{(A\mathbf{w} - \mathbf{b})}_{\nabla J(\mathbf{w})} < 0$

Interpretation of the gradient algorithm

For each iteration, minimize a local approximation of the problem.

Definition

A function J is gradient Lipschitz if there exist a constant L_J such that

$$\|\nabla J(\mathbf{w} + \mathbf{d}) - \nabla J(\mathbf{w})\| \leq L_J \|\mathbf{d}\|, \quad \forall \mathbf{d} \in \mathbb{R}^d, \forall \mathbf{w} \in \mathbb{R}^d. \quad (12)$$

Constante L_J is called the Lipschitz constant of ∇J .

Example

The least square loss $J_1(\mathbf{w})$ is gradient Lipschitz with constant

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

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

e

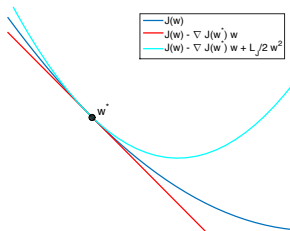
$$\|\nabla J_1(\mathbf{w} + \mathbf{d}) - \nabla J_1(\mathbf{w})\| \leq \|\mathbf{X}^\top \mathbf{X}\| \|\mathbf{d}\| = L_J \|\mathbf{d}\|.$$

Descent Lemma

[Ber99, Prop. A.24]

If J is gradient Lipschitz, we have the following approximation of J around point \mathbf{w} $\forall \mathbf{d} \in \mathbb{R}^d, \forall \mathbf{w} \in \mathbb{R}^d$.

$$J(\mathbf{w} + \mathbf{d}) \leq J(\mathbf{w}) + \nabla J(\mathbf{w})^\top \mathbf{d} + \frac{L_J}{2} \|\mathbf{d}\|^2$$



Sufficient conditions to prove convergence:

- $\mathbf{d}^{(k)} = -\frac{1}{L_J} \nabla J(\mathbf{w}^{(k)})$.
- et and a step size $\rho^{(k)} \leq \frac{1}{L_J}$

The Hessian

Assume J two times differentiable

Definition (Hessien)

Le Hessien $\nabla^2 J(\mathbf{w})$ d'une fonction J au point \mathbf{w} est la matrice de taille $d \times d$ de composantes $\nabla_{ij}^2 J(\mathbf{w}) = \frac{\partial^2 J}{\partial w_i \partial w_j}(\mathbf{w})$

Example

Le Hessien de J_1 le coût des moindres carrés de l'exemple (1) est

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

J est convexe si et seulement si

$\forall \mathbf{w} \in \mathbb{R}^d$, $\nabla^2 J(\mathbf{w})$ est une matrice définie positive

La méthode de Newton

utiliser l'approximation locale du second ordre suivante pour minimiser J :

$$J(\mathbf{w} + \mathbf{d}) = J(\mathbf{w}) + \mathbf{d}^\top \nabla J(\mathbf{w}) + \frac{1}{2} \mathbf{d}^\top \nabla^2 J(\mathbf{w}) \mathbf{d} + o(\|\mathbf{d}\|^2). \quad (13)$$

Definition (la méthode de Newton)

minimiser, à chaque itération, l'approximation quadratique de J

$$\mathbf{d}^{(k)} = -(\nabla^2 J)^{-1}(\mathbf{w}^{(k)}) \nabla J(\mathbf{w}^{(k)})$$

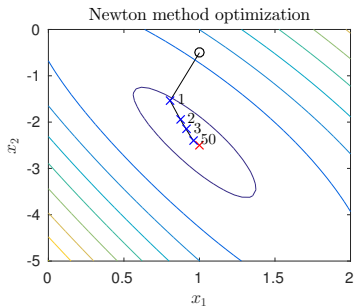
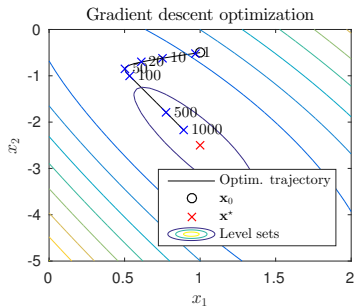
Exemple

Les itérations des méthodes de descente de gradient et de Newton pour le problème de minimisation des moindres carrés sont :

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \rho^{(k)} \mathbf{X}^\top (\mathbf{X} \mathbf{w}^{(k)} - \mathbf{Y}), && \text{Gradient} \\ \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{X} \mathbf{w}^{(k)} - \mathbf{Y}). && \text{Newton} \end{aligned}$$

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

Illustration de la méthode de newton



<i>méthode</i>	<i>direction de descente</i>	<i>temps de calcul</i>	<i>convergence</i>
gradient	$d = -\nabla J$	$\mathcal{O}(n)$	linéaire
quasi Newton	$d = -B\nabla J$	$\mathcal{O}(n^2)$	super linéaire
Newton	$d = -H^{-1}\nabla J$	$\mathcal{O}(n^3)$	quadratic

le temps de calcul du pas optimal peut aussi varier

si n est très grand, utiliser une approximation stochastique du gradient comme direction de descente [pour plus de détails voir par exemple Bot10]

Exemple : la régression logistique

La régression logistique binaire avec $Y \in \{0, 1\}^n$,

$$\min_{\mathbf{w} \in \mathbb{R}^p} J_\ell(\mathbf{w}) = \sum_{i=1}^n (-Y_i(\mathbf{X}\mathbf{w})_i + \log(1 + \exp(\mathbf{X}\mathbf{w})_i)), \quad (14)$$

avec $(\mathbf{X}\mathbf{w})_i$ la $i^{\text{ème}}$ composante du vecteur $\mathbf{X}\mathbf{w}$.

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 :

$$\begin{aligned} \nabla J_\ell(x) &= \mathbf{X}^\top (\mathbf{p} - Y) \\ \nabla^2 J_\ell(x) &= \mathbf{X}^\top \mathbf{P} \mathbf{X}, \end{aligned} \quad (15)$$

avec $\mathbf{p} \in \mathbb{R}^p$ le vecteur de composantes

$p_i(\mathbf{w}) = \exp((\mathbf{X}\mathbf{w})_i) / (1 + \exp((\mathbf{X}\mathbf{w})_i))$ et \mathbf{P} une matrice diagonale de terme général $\mathbf{P}_{ii} = p_i(1 - p_i)$, $i = 1, n$. 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 régression logistique : bricolage pratique

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

Algorithm 1 Méthode de Newton pour la régression logistique

- 1: **Entrées** : \mathbf{X} , \mathbf{Y} données d'apprentissage.
 - 2: **Sortie** : \mathbf{w} paramètres du modèle
 - 3: $\mathbf{w} \leftarrow 0$ initialisation des paramètres
 - 4: **while** on n'a pas convergé **do**
 - 5: $\mathbf{p} \leftarrow \frac{\exp^{\mathbf{X}\mathbf{w}}}{1 + \exp^{\mathbf{X}\mathbf{w}}}$ division terme à terme
 - 6: $P_{ii} \leftarrow p_i(1 - p_i), \quad i = 1, n$ équation (15)
 - 7: $\mathbf{z} \leftarrow \mathbf{X}\mathbf{w} + \mathbf{P}^{-1}(\mathbf{Y} - \mathbf{p})$
 - 8: $\mathbf{w} \leftarrow (\mathbf{X}^\top \mathbf{P} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{P} \mathbf{z}$
 - 9: **end while**
-

Plan

- 1 Examples of optimization for machine learning
- 2 Unconstrained convex and differentiable optimization
- 3 Constrained convex and differentiable optimization
- 4 Nondifferentiable unconstrained optimization

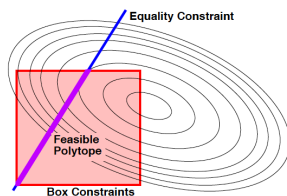
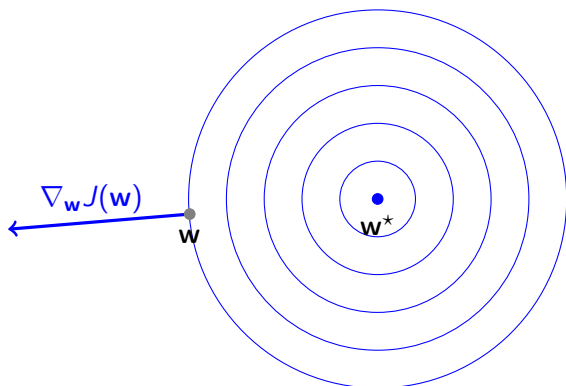


Figure from L. Bottou & C.J. Lin, Support vector machine solvers, in Large scale kernel machines, 2007.

Un exemple simple (pour commencer)

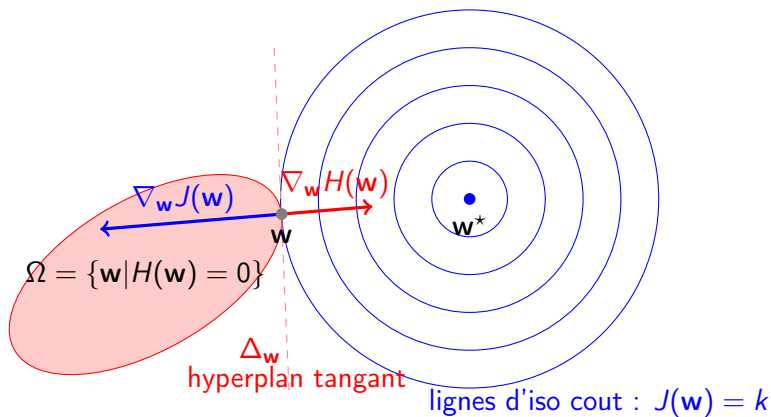
$$\begin{cases} \min_{w_1, w_2} & J(\mathbf{w}) = (w_1 - a)^2 + (w_2 - b)^2 \\ \text{avec} & \end{cases}$$



lignes d'iso cout : $J(\mathbf{w}) = k$

Un exemple simple (pour commencer)

$$\begin{cases} \min_{w_1, w_2} & J(\mathbf{w}) = (w_1 - a)^2 + (w_2 - b)^2 \\ \text{avec} & H(\mathbf{w}) = \alpha(w_1 - c)^2 + \beta(w_2 - d)^2 + \gamma w_1 w_2 - 1 \end{cases}$$



$$\nabla_{\mathbf{w}} H(\mathbf{w}) = \mu \nabla_{\mathbf{w}} J(\mathbf{w})$$

Le cas d'une seule contrainte d'égalité

$$\begin{cases} \min_{\mathbf{w}} & J(\mathbf{w}) & J(\mathbf{w} + \varepsilon \mathbf{d}) \approx J(\mathbf{w}) + \varepsilon \nabla_{\mathbf{w}} J(\mathbf{w})^T \mathbf{d} \\ \text{with} & H(\mathbf{w}) = 0 & H(\mathbf{w} + \varepsilon \mathbf{d}) \approx H(\mathbf{w}) + \varepsilon \nabla_{\mathbf{w}} H(\mathbf{w})^T \mathbf{d} \end{cases}$$

Cout J : \mathbf{d} est une direction de descente s'il existe $\varepsilon_0 \in \mathbb{R}$ tel que $\forall \varepsilon \in \mathbb{R}, 0 < \varepsilon \leq \varepsilon_0$

$$J(\mathbf{w} + \varepsilon \mathbf{d}) < J(\mathbf{w}) \quad \Rightarrow \quad \nabla_{\mathbf{w}} J(\mathbf{w})^T \mathbf{d} < 0$$

Contrainte H : \mathbf{d} est une direction de descente **admissible** s'il existe $\varepsilon_0 \in \mathbb{R}$ tel que $\forall \varepsilon \in \mathbb{R}, 0 < \varepsilon \leq \varepsilon_0$

$$H(\mathbf{w} + \varepsilon \mathbf{d}) = 0 \quad \Rightarrow \quad \nabla_{\mathbf{w}} H(\mathbf{w})^T \mathbf{d} = 0$$

si au point \mathbf{w}^* , les vecteurs $\nabla_{\mathbf{w}} J(\mathbf{w}^*)$ et $\nabla_{\mathbf{w}} H(\mathbf{w}^*)$ sont **colinéaires**, il n'existe pas de direction \mathbf{d} de descente admissible (sauf maximum).

→ \mathbf{w}^* est donc une solution locale du problème.

Multiplicateurs de Lagrange

Supposons que J et les fonctions H_i sont continument différentiables (et indépendantes).

$$\mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) \\ \text{avec} & H_1(\mathbf{w}) = 0 \\ \text{et} & H_2(\mathbf{w}) = 0 \\ & \dots \\ & H_p(\mathbf{w}) = 0 \end{cases}$$

Multiplicateurs de Lagrange

Supposons que J et les fonctions H_i sont continument différentiables (et indépendantes).

$$\mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) \\ \text{avec} & H_1(\mathbf{w}) = 0 & \mu_1 \\ \text{et} & H_2(\mathbf{w}) = 0 & \mu_2 \\ & \dots \\ & H_p(\mathbf{w}) = 0 & \mu_p \end{cases}$$

chaque contrainte est associée avec μ_i : un multiplicateurs de Lagrange.

Multiplicateurs de Lagrange

Supposons que J et les fonctions H_i sont continument différentiables (et indépendantes).

$$\mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) \\ \text{avec} & H_1(\mathbf{w}) = 0 & \mu_1 \\ \text{et} & H_2(\mathbf{w}) = 0 & \mu_2 \\ & \dots \\ & H_p(\mathbf{w}) = 0 & \mu_p \end{cases}$$

chaque contrainte est associée avec μ_i : un multiplicateurs de Lagrange.

Theorem (Conditions d'optimalité du premier ordre)

Pour qu'un point \mathbf{w} soit un extremum local le \mathcal{P} , il est nécessaire que :

$$\nabla_x J(\mathbf{w}^*) + \sum_{i=1}^p \mu_i \nabla_x H_i(\mathbf{w}^*) = 0 \quad \text{et} \quad H_i(\mathbf{w}^*) = 0, \quad i = 1, p$$

Un exemple où ça marche

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^3} & J(\mathbf{w}) = -w_1 w_2 - w_1 w_3 - w_2 w_3 \\ \text{avec} & H(\mathbf{w}) = w_1 + w_2 + w_3 - 3 = 0 \end{cases}$$

Un exemple où ça marche

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^3} & J(\mathbf{w}) = -w_1 w_2 - w_1 w_3 - w_2 w_3 \\ \text{avec} & H(\mathbf{w}) = w_1 + w_2 + w_3 - 3 = 0 \end{cases}$$

$$\nabla_x J(x) = - \begin{pmatrix} w_2 + w_3 \\ w_1 + w_3 \\ w_1 + w_2 \end{pmatrix} \quad \nabla_x H(x) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Les conditions d'optimalité sont

$$\begin{cases} -w_2 - w_3 + \mu = 0 \\ -w_1 - w_3 + \mu = 0 \\ -w_1 - w_2 + \mu = 0 \\ w_1 + w_2 + w_3 = 3 \end{cases}$$

la résolution du système $(A \setminus b)$ donne :

$$w_1 = w_2 = w_3 = 1 \quad \text{et} \quad \mu = 2$$

Un autre exemple où ça marche

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) = -\frac{1}{2} \mathbf{w}^\top A \mathbf{w} - \mathbf{w}^\top \mathbf{b} \\ \text{avec} & C \mathbf{w} = \mathbf{d} \end{cases}$$

$$\begin{aligned} \nabla_{\mathbf{w}} J &= A \mathbf{w} - \mathbf{b} \\ \nabla_{\mathbf{w}} H &= C^\top \end{aligned}$$

$$\begin{aligned} \nabla_x J(x) + \sum_{i=1}^p \mu_i \nabla_x H_i(\mathbf{w}) = 0 &\Rightarrow A \mathbf{w} + C^\top \boldsymbol{\mu} = \mathbf{b} \\ H(\mathbf{w}) = 0 &\Rightarrow C \mathbf{w} = \mathbf{d} \end{aligned}$$

...et on résout le système linéaire.

Attention si en plus on cherche $\mathbf{w} \geq 0$ ça devient plus compliqué

Un exemple où ça ne marche pas

$$\left\{ \begin{array}{l} \min_{w_1, w_2} \quad w_1 + w_2 \\ \text{avec} \quad (w_1 + 1)^2 + w_2^2 = 1 \\ \text{et} \quad (w_1 - 2)^2 + w_2^2 = 4 \end{array} \right.$$

le minimum est $(0, 0)$ l'unique solution réalisable ! Dans ce cas, il n'existe pas de multiplicateurs de Lagrange

lagrangien

Une fonction bien pratique :

définition : lagrangien

On appelle lagrangien du problème \mathcal{P} la fonction L définie par :

$$L(\mathbf{w}, \mu) = J(x) + \sum_{i=1}^p \mu_i H_i(\mathbf{w})$$

Grâce au lagrangien on retrouve les conditions d'optimalité :

$$\begin{cases} \nabla_x L(\mathbf{w}, \mu) = 0 & \Rightarrow \nabla_x J(\mathbf{w}) + \sum_{i=1}^p \mu_i \nabla_x H_i(\mathbf{w}) = 0 \\ \nabla_{\mu_i} L(\mathbf{w}, \mu) = 0 & \Rightarrow H_i(\mathbf{w}) = 0 \end{cases}$$

Interprétation graphique : optimisation multicritères.

Le cas d'une seule contrainte d'inégalité

$$\begin{cases} \min_{\mathbf{w}} & J(\mathbf{w}) & J(\mathbf{w} + \varepsilon \mathbf{d}) \approx J(\mathbf{w}) + \varepsilon \nabla_{\mathbf{w}} J(\mathbf{w})^T \mathbf{d} \\ \text{avec} & G(\mathbf{w}) \leq 0 & G(\mathbf{w} + \varepsilon \mathbf{d}) \approx G(\mathbf{w}) + \varepsilon \nabla_{\mathbf{w}} G(\mathbf{w})^T \mathbf{d} \end{cases}$$

Cout J : \mathbf{d} est une direction de descente, s'il existe un $\varepsilon_0 \in \mathbb{R}$ tel que $\forall \varepsilon \in \mathbb{R}, 0 < \varepsilon \leq \varepsilon_0$

$$J(\mathbf{w} + \varepsilon \mathbf{d}) < J(\mathbf{w}) \quad \Rightarrow \quad \nabla_{\mathbf{w}} J(\mathbf{w})^T \mathbf{d} < 0$$

Contrainte G : \mathbf{d} est une direction de descente admissible, s'il existe un $\varepsilon_0 \in \mathbb{R}$ tel que $\forall \varepsilon \in \mathbb{R}, 0 < \varepsilon \leq \varepsilon_0$

$$G(\mathbf{w} + \varepsilon \mathbf{d}) \leq 0 \quad \Rightarrow \quad \begin{array}{l} G(\mathbf{w}) < 0 : \text{no limit here on } \mathbf{d} \\ G(\mathbf{w}) = 0 : \nabla_{\mathbf{w}} G(\mathbf{w})^T \mathbf{d} \leq 0 \end{array}$$

Les deux alternatives pour caractériser l'optimalité

Si x^* se trouve sur la limite du domaine admissible ($G(\mathbf{w}^*) = 0$), alors si les vecteurs $\nabla_{\mathbf{w}} J(\mathbf{w}^*)$ et $\nabla_{\mathbf{w}} G(\mathbf{w}^*)$ sont colinéaires **et dans des directions opposées**, il n'existe pas de direction de descente admissible \mathbf{d} à ce point. x^* est donc un optimum local... Ou alors $\nabla_{\mathbf{w}} J(\mathbf{w}^*) = 0$

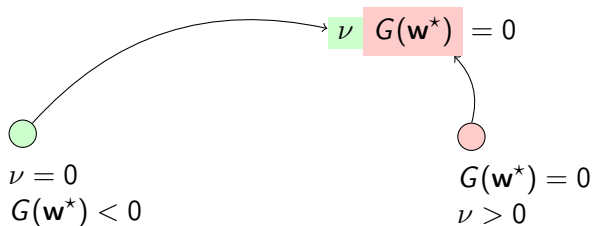
Les deux alternatives à l'optimum

$$\nabla_{\mathbf{w}} J(\mathbf{w}^*) = -\nu \nabla_{\mathbf{w}} G(\mathbf{w}^*) \quad \text{et} \quad \nu > 0; G(\mathbf{w}^*) = 0$$

or

$$\nabla_{\mathbf{w}} J(\mathbf{w}^*) = 0 \quad \text{et} \quad \nu = 0; G(\mathbf{w}^*) < 0$$

Cette alternative est résumée par la condition de complémentarité



Conditions d'optimalité du premier ordre (1)

$$\text{problem } \mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^d} & J(\mathbf{w}) \\ \text{avec} & h_j(\mathbf{w}) = 0 \quad j = 1, \dots, \ell \\ \text{et} & g_i(\mathbf{w}) \leq 0 \quad i = 1, \dots, q \end{cases}$$

Definition (Conditions de Karush, Kuhn and Tucker)

stationarité $\nabla J(\mathbf{w}^*) + \sum_{j=1}^{\ell} \mu_j \nabla h_j(\mathbf{w}^*) + \sum_{i=1}^q \nu_i \nabla g_i(\mathbf{w}^*) = 0$

admissibilité primale $h_j(\mathbf{w}^*) = 0 \quad j = 1, \dots, \ell$
 $g_i(\mathbf{w}^*) \leq 0 \quad i = 1, \dots, q$

admissibilité duale $\nu_i \geq 0 \quad i = 1, \dots, q$

complémentarité $\nu_i g_i(\mathbf{w}^*) = 0 \quad i = 1, \dots, q$

μ_j et ν_i sont les multiplicateurs de Lagrange du problème \mathcal{P}

Exemple (Les conditions KKT dans un exemple simple)

Considérons le problème d'optimisation en une dimension :

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

admet les conditions KKT suivantes :

stationnarité	$(w + 1) - \nu = 0$
admissibilité primale	$-w \leq 0$
admissibilité duale	$\nu \geq 0$
complémentarité	$\nu w = 0.$

La condition de complémentarité impose que w ou ν soit nul. S'il s'agit de ν , 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 (Moindres carrés positifs)

Il s'agit d'un cas particulier du problème (9) avec le coût des moindres carrés et sans pénalisation. Le problème s'écrit alors :

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{Y}\|^2 \\ \text{avec} & 0 \leq w_i, \quad i = 1, \dots, p \end{cases}$$

Les conditions KKT de ce problème sont :

stationnarité	$\mathbf{X}^t(\mathbf{X}\mathbf{w} - \mathbf{Y}) - \boldsymbol{\nu} = \mathbf{0}$
admissibilité primale	$-\mathbf{w} \leq \mathbf{0}$
admissibilité duale	$\boldsymbol{\nu} \geq \mathbf{0}$
complémentarité	$\text{diag}(\boldsymbol{\nu})\mathbf{w} = \mathbf{0}$.

Conditions d'optimalité du premier ordre (2)

Théorème (12.1 Nocedal & Wright pp 321)

Si le vecteur x^* est un point stationnaire du problème \mathcal{P}

Alors il existe ^a des multiplicateurs de Lagrange $(w^*, \{\mu_j\}_{j=1:\ell}, \{\nu_i\}_{i=1:q})$ vérifiant les conditions KKT

^a sous certaines conditions sur les contraintes, par exemple leur indépendance

Si le problème est **convexe**, alors tout point stationnaire est solution du problème

Conditions KKT - Lagrangien (3)

$$\text{problem } \mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) \\ \text{avec} & h_j(\mathbf{w}) = 0 \quad j = 1, \dots, p \\ \text{et} & g_i(\mathbf{w}) \leq 0 \quad i = 1, \dots, q \end{cases}$$

Définition : Lagrangien

Le lagrangien du problème \mathcal{P} est la fonction suivante :

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

L'importance du lagrangien

- la condition de stationnarité est donnée par : $\nabla \mathcal{L}(\mathbf{w}^*, \mu, \nu) = 0$
- l'optimum est un point selle de lagrangien $\max_{\mu, \nu} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mu, \nu)$

Variables **primales** : \mathbf{x} et variables **duales** μ, ν (Les multiplicateurs de Lagrange)

Dualité – définitions (1)

Exemple (Lagrangien de l'exemple 18)

Il s'agit d'un cas particulier du problème (9) avec le coût des moindres carrés et sans pénalisation. Le problème s'écrit alors :

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & \frac{1}{2} \|\mathbf{X}\mathbf{w} - Y\|^2 \\ \text{avec} & 0 \leq w_i, \quad i = 1, \dots, p \end{cases}$$

Le Lagrangien est :

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

Exemple (L'exemple 2 et le lagrangien)

L et Ω convexes

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^d} & L(\mathbf{w}) \\ \text{avec} & \Omega(\mathbf{w}) \leq k, \end{cases} \quad (16)$$

Le lagrangien de ce problème est

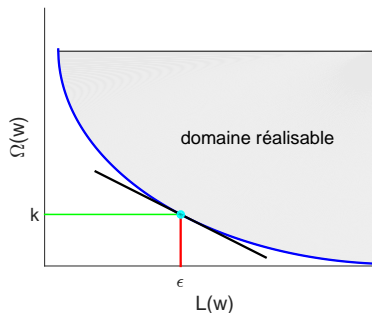
$$\mathcal{L}(\mathbf{w}, \nu) = L(\mathbf{w}) + \nu(\Omega(\mathbf{w}) - k),$$

pour un k donné, il existe un ν

$$\min_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) + \nu \Omega(\mathbf{w}) \quad (17)$$

Ce même raisonnement peut être formulé à partir de problème suivant :

$$\begin{cases} \min_{\mathbf{w} \in \mathbb{R}^d} & \Omega(\mathbf{w}) \\ \text{avec} & L(\mathbf{w}) \leq \epsilon, \end{cases} \quad (18)$$



Dualité – définitions (1)

Problème primal et son dual lagrangien

$$\mathcal{P} = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^n} & J(\mathbf{w}) \\ \text{avec} & h_j(x) = 0 \quad j = 1, \ell \\ \text{et} & g_i(x) \leq 0 \quad i = 1, q \end{cases} \quad \mathcal{D} = \begin{cases} \max_{\mu \in \mathbb{R}^p, \nu \in \mathbb{R}^q} & Q(\mu, \nu) \\ \text{avec} & \nu_j \geq 0 \quad j = 1, q \end{cases}$$

La fonction objectif duale :

$$\begin{aligned} Q(\mu, \nu) &= \inf_x \mathcal{L}(\mathbf{w}, \mu, \nu) \\ &= \inf_x J(x) + \sum_{j=1}^p \mu_j h_j(x) + \sum_{i=1}^q \nu_i g_i(x) \end{aligned}$$

Le dual de Wolfe

$$\mathcal{W} = \begin{cases} \max_{\mathbf{w}, \mu \in \mathbb{R}^p, \nu \in \mathbb{R}^q} & \mathcal{L}(\mathbf{w}, \mu, \nu) \\ \text{avec} & \nu_j \geq 0 \quad j = 1, q \\ \text{et} & \nabla J(x) + \sum_{j=1}^p \mu_j \nabla h_j(x) + \sum_{i=1}^q \nu_i \nabla g_i(x) = 0 \end{cases}$$

Exemple (Calcul du problème dual de l'exemple 18)

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

Le gradient du lagrangien par rapport à \mathbf{w} est $\mathbf{X}^t(\mathbf{X}\mathbf{w} - Y) - \nu$ ce qui nous donne $\mathbf{w}^* = (\mathbf{X}^t\mathbf{X})^{-1}(\nu + \mathbf{X}^t Y)$.

$$Q(\nu) = -\frac{1}{2}(\mathbf{X}^t Y + \nu)(\mathbf{X}^t\mathbf{X})^{-1}(\mathbf{X}^t Y + \nu)$$

et le problème dual est

$$\begin{cases} \max_{\nu \in \mathbb{R}^q} & -\frac{1}{2}\nu^t(\mathbf{X}^t\mathbf{X})^{-1}\nu - Y^t\mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\nu \\ \text{avec} & \nu_j \geq 0, \quad j = 1, \dots, q, \end{cases}$$

la condition de stationnarité de l'exemple (18) $\nu = \mathbf{X}^t(\mathbf{X}\mathbf{w} - Y)$, le problème dual peut être reformulé à travers les variables primales comme :

$$\begin{cases} \max_{\mathbf{w} \in \mathbb{R}^d} & -\frac{1}{2}\mathbf{w}^t\mathbf{X}^t\mathbf{X}\mathbf{w} \\ \text{avec} & \mathbf{X}^t(\mathbf{X}\mathbf{w} - Y) \geq 0. \end{cases}$$

Dualité – théorèmes (2)

Théorème (12.12, 12.13 and 12.14 Nocedal & Wright pp 346)

Si J , et g_i sont convexes et continument différentiables et les h_j affines, ^a, alors les solutions des problèmes primale et dual ont le même coût

^a sous certaines conditions sur les contraintes, par exemple leur indépendance

$$\begin{aligned}(\mu^*, \nu^*) &= \text{solution du problème } \mathcal{D} \\ \mathbf{w}^* &= \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mu^*, \nu^*)\end{aligned}$$

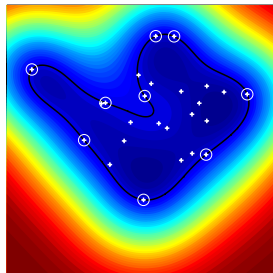
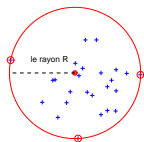
$$\begin{aligned}Q(\mu^*, \nu^*) &= \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mu^*, \nu^*) = \mathcal{L}(\mathbf{w}^*, \mu^*, \nu^*) \\ &= J(\mathbf{w}^*) + \mu^* H(\mathbf{w}^*) + \nu^* G(\mathbf{w}^*) = J(\mathbf{w}^*)\end{aligned}$$

pour tout point \mathbf{w} admissible

$$Q(\mu, \nu) \leq J(\mathbf{w}) \quad \rightarrow \quad 0 \leq J(\mathbf{w}) - Q(\mu, \nu)$$

le **saut de dualité** est la différence entre les coûts primal et dual

Exemple : la boule minimum englobante ou SVDD



Étant donné un ensemble de n observations $\{X_i \in \mathbb{R}^p, i = 1, \dots, n\}$ ce problème consiste à trouver la boule p dimensionnelle de centre \mathbf{c} de rayon minimum R contenant tous les points X_i .

$$\begin{cases} \min_{R \in \mathbb{R}, \mathbf{c} \in \mathbb{R}^p} & R^2 \\ \text{avec} & \|X_i - \mathbf{c}\|^2 \leq R^2, \quad i = 1, \dots, n. \end{cases} \quad (19)$$

Les KKT des SVDD

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

KKT conditionns :

stationarté $\triangleright 2\mathbf{c} \sum_{i=1}^n \nu_i - 2 \sum_{i=1}^n \nu_i X_i = 0 \quad \leftarrow$ théorème de représentation

$\triangleright 1 - \sum_{i=1}^n \nu_i = 0$

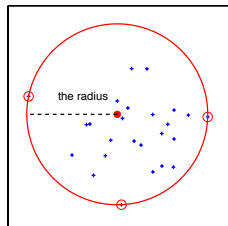
primal admiss. $\|X_i - \mathbf{c}\|^2 \leq R^2$

dual admiss. $\nu_i \geq 0 \quad i = 1, n$

complementarité $\nu_i (\|X_i - \mathbf{c}\|^2 - R^2) = 0 \quad i = 1, n$

Les KKT des SVDD

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



KKT conditionns :

stationarté $\triangleright 2\mathbf{c} \sum_{i=1}^n \nu_i - 2 \sum_{i=1}^n \nu_i X_i = 0 \quad \leftarrow$ théorème de représentation

$$\triangleright 1 - \sum_{i=1}^n \nu_i = 0$$

primal admiss. $\|X_i - \mathbf{c}\|^2 \leq R^2$

dual admiss. $\nu_i \geq 0 \quad i = 1, n$

complementarité $\nu_i (\|X_i - \mathbf{c}\|^2 - R^2) = 0 \quad i = 1, n$

La complémentarité définit deux groupes de points

les vecteurs support $\|X_i - \mathbf{c}\|^2 = R^2$ et les points de l'intérieur $\nu_i = 0$

MEB: Dual

Le théorème de représentation

$$\mathbf{c} = \frac{\sum_{i=1}^n \nu_i X_i}{\sum_{i=1}^n \nu_i} = \sum_{i=1}^n \nu_i X_i$$

$$\mathcal{L}(\nu) = \sum_{i=1}^n \nu_i \left(\|X_i - \sum_{j=1}^n \nu_j \mathbf{w}_j\|^2 \right)$$

$$\sum_{i=1}^n \sum_{j=1}^n \nu_i \nu_j X_i^\top X_j = \nu^\top G \nu \quad \text{and} \quad \sum_{i=1}^n \nu_i X_i^\top X_i = \nu^\top \text{diag}(G)$$

avec $G = XX^\top$ la matrice de Gram : $G_{ij} = X_i^\top X_j$,

$$\left\{ \begin{array}{l} \min_{\nu \in \mathbb{R}^n} \quad \nu^\top G \nu - \nu^\top \text{diag}(G) \\ \text{avec} \quad e^\top \nu = 1 \\ \text{et} \quad 0 \leq \nu_i, \end{array} \right. \quad i = 1 \dots n$$

MEB: Dual

Le théorème de représentation

$$\mathbf{c} = \frac{\sum_{i=1}^n \nu_i X_i}{\sum_{i=1}^n \nu_i} = \sum_{i=1}^n \nu_i X_i$$

$$\mathcal{L}(\nu) = \sum_{i=1}^n \nu_i \left(\|X_i - \sum_{j=1}^n \nu_j \mathbf{w}_j\|^2 \right)$$

$$\sum_{i=1}^n \sum_{j=1}^n \nu_i \nu_j X_i^\top X_j = \nu^\top G \nu \quad \text{and} \quad \sum_{i=1}^n \nu_i X_i^\top X_i = \nu^\top \text{diag}(G)$$

avec $G = XX^\top$ la matrice de Gram : $G_{ij} = X_i^\top X_j$,

$$\left\{ \begin{array}{l} \min_{\nu \in \mathbb{R}^n} \quad \nu^\top G \nu - \nu^\top \text{diag}(G) \\ \text{avec} \quad e^\top \nu = 1 \\ \text{et} \quad 0 \leq \nu_i, \end{array} \right. \quad i = 1 \dots n$$

SVDD primal vs. dual

Primal

$$\left\{ \begin{array}{l} \min_{R \in \mathbb{R}, \mathbf{c} \in \mathbb{R}^p} R^2 \\ \text{avec} \quad \|X_i - \mathbf{c}\|^2 \leq R^2, \\ \quad \quad \quad i = 1, \dots, n \end{array} \right.$$

- $p + 1$ inconnues
- n contraintes
- peut être reformulé QP
- bien quand $d \ll n$

Dual

$$\left\{ \begin{array}{l} \min_{\nu} \nu^T G \nu - \nu^T \text{diag}(G) \\ \text{avec} \quad e^T \nu = 1 \\ \text{et} \quad 0 \leq \nu_i, \\ \quad \quad \quad i = 1 \dots n \end{array} \right.$$

- n inconnues avec G la matrice des influences des couples
- n contraintes de boîte
- moins difficiles à résoudre
- à utiliser lorsque $d > n$

SVDD primal vs. dual

Primal

$$\left\{ \begin{array}{l} \min_{R \in \mathbb{R}, c \in \mathbb{R}^p} R^2 \\ \text{avec} \quad \|X_i - c\|^2 \leq R^2, \\ \quad \quad i = 1, \dots, n \end{array} \right.$$

- $p + 1$ inconnues
- n contraintes
- peut être reformulé QP
- bien quand $d \ll n$

Dual

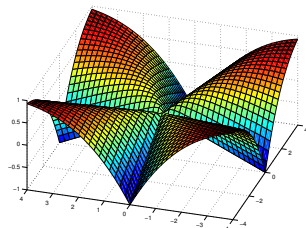
$$\left\{ \begin{array}{l} \min_{\nu} \nu^T G \nu - \nu^T \text{diag}(G) \\ \text{avec} \quad e^T \nu = 1 \\ \text{et} \quad 0 \leq \nu_i, \\ \quad \quad i = 1 \dots n \end{array} \right.$$

- n inconnues avec G la matrice des influences des couples
- n contraintes de boîte
- moins difficiles à résoudre
- à utiliser lorsque $d > n$

Et ce sont tous les deux des QP convexes

Plan

- 1 Examples of optimization for machine learning
- 2 Unconstrained convex and differentiable optimization
- 3 Constrained convex and differentiable optimization
- 4 Nondifferentiable unconstrained optimization

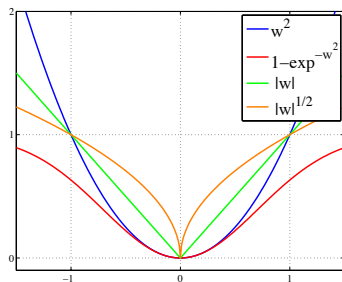


Formulation des problèmes d'optimisation non différentiable

Exemple (Les pénalités décomposables)

$$\min_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) + \lambda \Omega(\mathbf{w}), \quad \Omega(\mathbf{w}) = \sum_{i=1}^d \omega(w_i).$$

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



Le cas non différentiable en zéro

L'approche proximale

Supposons $L(\mathbf{w})$ Lipschitz gradient, le *descent lemma* (23) stipule

$$J(\mathbf{w}) \leq L(\mathbf{w}^{(k)}) + \nabla L(\mathbf{w}^{(k)})^\top (\mathbf{w} - \mathbf{w}^{(k)}) + \frac{1}{2\rho} \|\mathbf{w} - \mathbf{w}^{(k)}\|^2 + \lambda\Omega(\mathbf{w}),$$

$\rho \leq \frac{1}{L_L}$ avec L_L la constante de Lipschitz de la fonction $L(\mathbf{w})$.

Minimisation de cette majoration

$$\min_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}^{(k)}) + \nabla L(\mathbf{w}^{(k)})^\top (\mathbf{w} - \mathbf{w}^{(k)}) + \frac{1}{2\rho} \|\mathbf{w} - \mathbf{w}^{(k)}\|^2 + \lambda\Omega(\mathbf{w}),$$

est équivalent à

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

avec $\mathbf{u} = \mathbf{w}^{(k)} - \rho \nabla L(\mathbf{w}^{(k)})$.

L'opérateur proximal

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

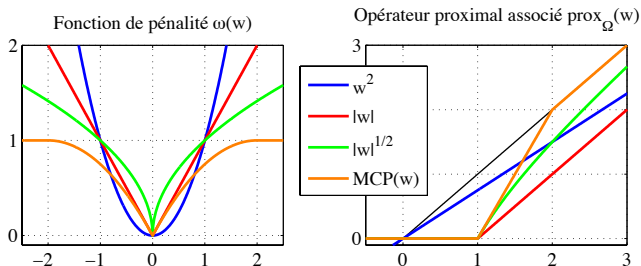
L'expression ci-dessus est associée à l'opérateur proximal :

Definition (Opérateur proximal)

L'opérateur proximal de la fonction Ω est :

$$\begin{aligned} \text{prox}_{\Omega} : \mathbb{R}^d &\longrightarrow \mathbb{R}^d \\ \mathbf{w} &\longmapsto \text{prox}_{\Omega}(\mathbf{w}) = \arg \min_{\mathbf{u} \in \mathbb{R}^d} \Omega(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2. \end{aligned}$$

$\Omega(\mathbf{w})$ est séparable



Exemple (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
$\Omega(\mathbf{w}) = \lambda \ \mathbf{w}\ _1$	$\text{prox}_\Omega(\mathbf{w}) = s_w \max(0, \mathbf{w} - \lambda)$	seuillage doux
$\Omega(\mathbf{w}) = \lambda \ \mathbf{w}\ _{1/2}^{1/2}$	[XCXZ12, Equation 11]	<i>bridge or power family</i>
$\Omega(\mathbf{w}) = \mathbb{I}_C(\mathbf{w})$	$\text{prox}_\Omega(\mathbf{w}) = \arg \min_{\mathbf{u} \in C} \frac{1}{2} \ \mathbf{u} - \mathbf{w}\ ^2$	projection

Algorithme de descente de gradient proximal

Definition (Algorithme de descente de gradient proximal)

Un principe général pour résoudre le problème de MRE pénalisé consiste à mettre en œuvre les itérations suivantes

$$\mathbf{w}^{k+1} = \text{prox}_{\rho^{(k)}\lambda\Omega}(\mathbf{w}^k - \rho^{(k)}\nabla L(\mathbf{w}^k)).$$

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.

- Si le pas ρ est bien choisi, il converge vers un minimum
- utiliser la règle de Barzilai-Borwein [WNF09].
- il est aussi possible d'accélérer l'algorithme [N⁺07, BT09].
- code Matlab : github.com/rflamary/nonconvex-optimization

L'exemple du LASSO

$$L(\mathbf{w}) = \frac{1}{2} \|\mathbf{X}\mathbf{w} - Y\|^2 \text{ et } \Omega(\mathbf{w}) = \sum_{i=1}^p |w_i|$$

$$\begin{aligned} \text{prox}_{\Omega}(\mathbf{w}) &= \arg \min_{\mathbf{u} \in \mathbb{R}^n} \sum_{i=1}^p |u_i| + \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 \\ &= \arg \min_{\mathbf{u} \in \mathbb{R}^n} \sum_{i=1}^p (|u_i| + \frac{1}{2}(u_i - w_i)^2). \end{aligned}$$

Calcul de sa sous différentielle

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

de sorte que

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

la méthode du gradient proximal pour le LASSO

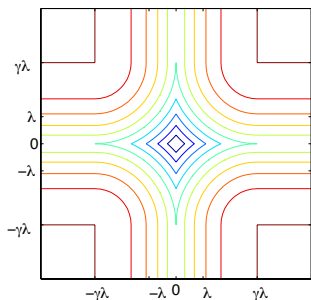
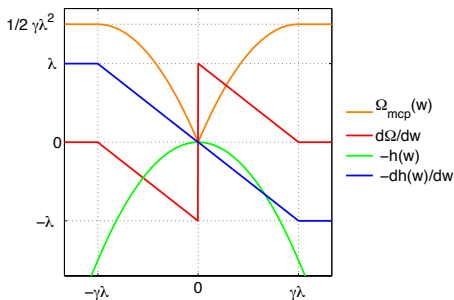
Algorithm 2 L'algorithme de gradient proximal pour le LASSO

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

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

La pénalité de type *minimax concave penalty* (MCP) pour ($\lambda \geq 0, \gamma \geq 1$)

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



MCP avec le cout logistique J_ℓ

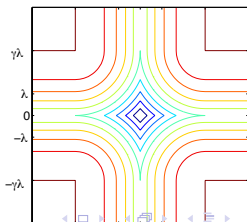
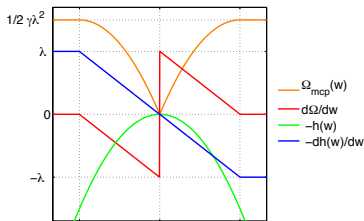
$$\min_{\mathbf{w} \in \mathbb{R}^p} \sum_{i=1}^n \left(-Y_i(\mathbf{X}\mathbf{w})_i + \log(1 + \exp(\mathbf{X}\mathbf{w})_i) \right) + \sum_{i=1}^p \Omega_{\lambda, \gamma}(|w_i|), \quad (20)$$

décomposition de la pénalité MCP

$$\min_{\mathbf{w} \in \mathbb{R}^p} \underbrace{J_\ell(\mathbf{w}) - \lambda \sum_{j=1}^p h_{\lambda, \gamma}(|w_j|)}_{J_m(\mathbf{w})} + \lambda \|\mathbf{w}\|_1, \quad (21)$$

avec

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

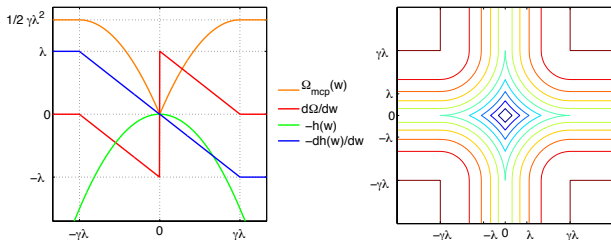


$J_m(\mathbf{w})$ est non convexe mais différentiable. Le gradient de J_m est

$$\nabla_{\beta} J_m(\mathbf{w}) = \mathbf{X}^{\top}(\mathbf{p} - Y) - \begin{cases} \text{sign}(w_j)\lambda & \text{si } |w_j| > \lambda\gamma \\ \frac{w_j}{\gamma} & \text{sinon.} \end{cases}$$

L'opérateur proximal de la norme ℓ_1 est

$$\text{prox}_{\ell_1}(u) = \begin{cases} 0 & \text{si } |u| \leq \lambda \\ \text{sign}(u)(|u| - \lambda) & \text{sinon.} \end{cases}$$



L'algorithme proximal ℓ_1 pour la régression logistique MCP

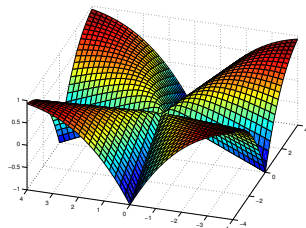
Algorithm 3 L'algorithme proximal ℓ_1 pour la régression logistique MCP

- 1: **Entrées** : \mathbf{X}, Y données d'apprentissage.
 - 2: **Sortie** : \mathbf{w} paramètres du modèle
 - 3: $\rho \leftarrow 1/\|\mathbf{X}^\top \mathbf{X}\|$ initialisation du pas
 - 4: **while** on n'a pas convergé **do**
 - 5: $\mathbf{p} \leftarrow \frac{\exp^{\mathbf{X}\mathbf{w}}}{1 + \exp^{\mathbf{X}\mathbf{w}}}$ division terme à terme
 - 6: $\mathbf{w} \leftarrow \mathbf{w} - \rho(\mathbf{X}^\top (\mathbf{p} - Y) - \text{sign}(\mathbf{w}) \min(\lambda, \frac{|\mathbf{w}|}{\gamma}))$
 - 7: $\mathbf{w} \leftarrow \text{sign}(\mathbf{w}) \max(0, |\mathbf{w}| - \rho\lambda)$ l'opérateur prox_{ℓ_1}
 - 8: **end while**
-

Soulignons enfin que la décomposition (21) n'est pas unique et qu'il est donc possible de dériver d'autres algorithmes proximaux pour résoudre ce problème.

Plan

- 1 Examples of optimization for machine learning
- 2 Unconstrained convex and differentiable optimization
- 3 Constrained convex and differentiable optimization
- 4 Nondifferentiable unconstrained optimization



Conclusion

- une certaine maturité de l'optimisation
- l'exemple du LASSO : la taille et la vitesse
 - ▶ LASSO avec CVX
 - ▶ le LASSO comme un QP
 - ▶ le LASSO à l'aide d'un algorithme proximal
- convexité et différentiabilité
- Le futur de l'optimisation
 - ▶ gros volumes de données : les méthodes stochastiques
 - ▶ non convexe, non différentiable

de 1996 à 2016

	facteur d'accélération
machine	$\times 2^{10} = 1000 - 1600$
solver	$\times 1000 - 3600$
formulation	???
global	$\times 1 - 5 \cdot 10^6$

Bibliography

- [Ber99] Dimitri P Bertsekas. Nonlinear programming. 1999.
- [Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
- [BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [GB14] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [HU02] Jean-Baptiste Hiriart-Urruty. L'optimisation: deux ou trois choses que je sais d'elle. *Note Technique- Centre national d'études spatiales*, 2002.
- [N⁺07] Yurii Nesterov et al. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.
- [NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [WNF09] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.
- [XCXZ12] Zongben Xu, Xiangyu Chang, Fengmin Xu, and Hai Zhang. $L_{1/2}$ regularization: a thresholding representation theory and a fast solver. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1013–1027, 2012.