Classification in High Dimension

Tristan Mary-Huard
maryhuar@agroparistech.fr

Génétique Quantitative et Évolution - Le Moulon INRA/Univ. Paris Sud/CNRS/AgroParisTech
AgroParisTech/INRA UMR 518

Overview

Prerequisites

"You know nothing, John Snow."

~~V.Vapnik~~
~~V.Koltchinskii~~
Traditional wildling saying

Overview

Basics in optimization

I - Theoretical aspects

An Introduction to Optimization [CZ13]
Convex Optimization [BV04]
(a.k.a. the convex surrogate of the Bible)

Standard optimization problem

Standard problem

$$\min_{x \in \Omega} \; f(x)$$

with $f : \mathbb{R}^p \rightarrow \mathbb{R}$ differentiable, and $\Omega \subset \mathbb{R}^p$.

Definition

x^* is a local minimizer iff

$$\exists \varepsilon / \forall x \in B(x^*, \varepsilon) \cap \Omega, \; f(x) \geq f(x^*)$$

x^* is a global minimizer iff

$$\forall x \in \Omega, \; f(x) \geq f(x^*)$$

First order necessary condition

Admissible direction

$d \in \mathbb{R}^p$ is admissible at point x if

$$\exists \alpha_0 > 0 / \forall \alpha \in [0, \alpha_0], x + \alpha d \in \Omega.$$

The directional derivative w.r.t. d is defined as

$$\frac{\partial f(x)}{\partial d} = \lim_{\alpha \to 0} \frac{f(x + \alpha d) - f(x)}{\alpha} = d^T \nabla f(x)$$

Theorem (1st order necessary condition)

If f is C^1 and x^* is a local minimizer of f over Ω. Then for all d admissible at point x^*,

$$d^T \nabla f(x^*) \geq 0$$

Note : If x^* is an interior point of Ω, then NC$\Rightarrow \nabla f(x^*) = 0$.

Convex optimization problems

Convex set, convex function

Ω is convex if $\forall (x, y, \lambda) \in \Omega^2 \times [0, 1]$,

$$\lambda x + (1 - \lambda) y \in \Omega$$

f is convex if $\forall (x, y, \lambda) \in \mathbb{R}^p \times \mathbb{R}^p \times [0, 1]$,

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

Proposition

- If f is convex, any local minimizer is a global minimizer.
- If f is convex and differentiable,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \ , \ \forall x, y$$

Convex Optimization problems

Standard problem

$$\min_{x \in \Omega} \; f(x)$$

with $f : \mathbb{R}^p \to \mathbb{R}$ differentiable, and $\Omega \in \mathbb{R}^p$.

Theorem

Assume f is convex and differentiable, and Ω is convex. Then $x^* \in \Omega$ is a global minimizer iff

$$< \langle \nabla f(x^*), y - x^* \rangle \geq 0, \; \forall y$$

Primal optimization problem

Consider problem

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0,\ \forall i = 1, ..., m \end{array}$$

New objective function :

$$\begin{aligned} f(x) + \sum_{i=1}^{m} \max_{\lambda_i \geq 0} \lambda_i g_i(x) &= \max_{\lambda \geq 0} \left\{ f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \right\} \\ &= \max_{\lambda \geq 0} L(x, \lambda) \end{aligned}$$

$\lambda_1, ..., \lambda_m$: Lagrange multipliers,
$L(.,.)$: Lagrange function.

The initial optimization problem becomes

$$\min_{x} \max_{\lambda \geq 0} L(x, \lambda) \qquad (\mathscr{P})$$

Dual optimization problem

Alternatively, consider problem

$$\max_{\lambda \geq 0} \min_{x} L(x, \lambda) \qquad (\mathscr{D})$$

$(\mathscr{D})$ is the *dual* problem associated with *primal* problem $(\mathscr{P})$.

Note $G(.)$ the dual function

$$G(\lambda) = \min_{x} L(x, \lambda)$$

Proposition

For all $\lambda \geq 0$, one has

$$G(\lambda) \leq p^*$$

where $p^* = f(x^*)$

Duality gap

Definition

Note $d^* = \max_{\lambda \geq 0} G(\lambda)$ *the solution of* $(\mathscr{D})$. *Then*

$$p^* - d^* \geq 0$$

is called the duality gap.
If $p^* - d^* = 0$, *then we say that **strong duality holds**.*

Questions

- How does strong duality help ?
- When does strong duality hold ?

Complementary slackness conditions

Proposition

If strong duality holds, then

$$\lambda_i^* g_i(x^*) = 0, \quad \forall i = 1, ..., m$$

where $\lambda^* = \arg\max_{\lambda \geq 0} \; G(\lambda)$.

Also note that x^* is the minimizer of $L(x, \lambda^*)$, therefore

$$\nabla L(x^*, \lambda^*) = 0$$

Karush Kuhn Tucker conditions

Proposition

If strong duality holds, the optimal Lagrange multiplier vector λ^* and the optimal solution x^* of $(\mathscr{P})$ satisfy

$$
\begin{aligned}
g_i(x^*) &\le 0, \quad \forall i = 1, ..., m && \textit{(primal feasability)}\\
\lambda_i^* &\ge 0, \quad \forall i = 1, ..., m && \textit{(dual feasability)}\\
\lambda_i^* g_i(x^*) &= 0, \quad \forall i = 1, ..., m && \textit{(compl. slackness)}\\
\nabla L(x^*, \lambda^*) = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) &= 0, && \textit{(first order condition)}
\end{aligned}
$$

Strong duality does not hold in general, but holds under mild conditions for convex optimization problems...

Slater's constraint qualification

Proposition

Consider problem $(\mathscr{P})$ where $f, g_1, ..., g_m$ are convex functions. Then strong duality holds if there exists a strictly feasible point, satisfying

$$g_i(x) < 0, \quad \forall i = 1, ..., m$$

Proof : Technical !
See [BV04]

Proposition

Assume $(\mathscr{P})$ is convex. Then if (λ^*, x^*) satisfy the KKT conditions, strong duality holds and (λ^*, x^*) is optimal.

So far...

Convex + differentiability

If $f, g_1, ..., g_m$ are differentiable and convex, then the KKT conditions are necessary and sufficient for optimality.

Potential use

⋆ Solve analytically the KKT conditions,
⋆ Guidelines for the development of efficient algorithms,
⋆ Solve the dual rather than the primal when easier !

Limitation

Some objective functions (hinge loss) and/or constraints (L_1 norm) are convex but not differentiable...

Subdifferential and subgradients

Recall that for a convex, differentiable function f

$$\forall x, y \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

Definition

Let $f : \mathbb{R}^p \to \mathbb{R}$. ω_x is a subgradient of f at point x if

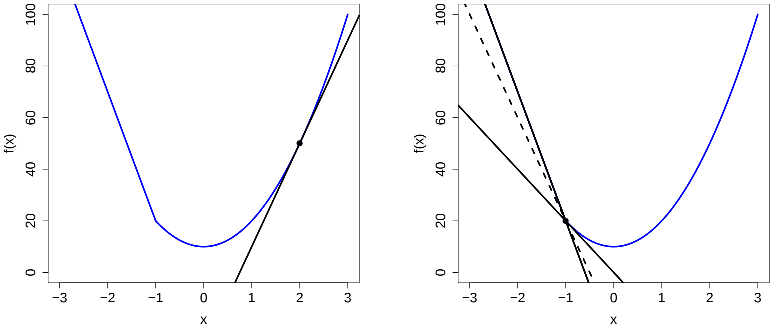
$$\forall x, y \quad f(y) \geq f(x) + \langle \omega_x, y - x \rangle$$

The set

$$\partial f(x) = \{\omega \ / \ \forall y \quad f(y) \geq f(x) + \langle \omega, y - x \rangle\}$$

is called the subdifferential of f at point x

A graphical illustration



At $x = 2$ the function is differentiable

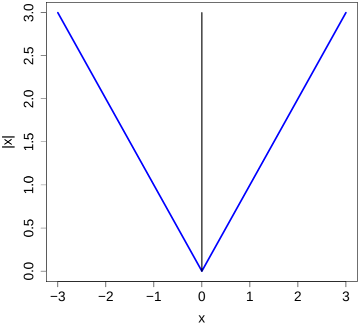
$\Rightarrow$ a unique tangent hyperplane

At $x = -1$ the function is not differentiable

$\Rightarrow$ many "lower" hyperplanes !

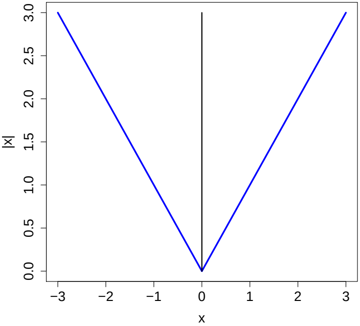
Subdifferential for the L_1 norm

$$\partial |x| \quad =$$



Subdifferential for the L_1 norm

$$\partial |x| = \begin{cases} \text{sign}[x] & \text{if } x \neq 0, \\ [-1,1] & \text{if } x = 0. \end{cases}$$



$$\partial ||x||_1 = \{\omega \in \mathbb{R}^p / \ \omega_j = \text{sign}[x_j] \text{ if } x_j \neq 0, \ \omega_j \in [-1,1] \text{ if } \ x_j = 0\}$$

Subdifferential and subgradients

Subdifferential and convexity

⋆ f is convex ⇒ $\partial f(x)$ is non-empty, $\forall x$,

⋆ f is convex and differentiable at x ⇒ $\partial f(x) = \{\nabla f(x)\}$.

Proof : See [Gir14]

Theorem

Assume f is convex and non-differentiable. Then

$$x^* = \arg\min_x \; f(x) \Leftrightarrow 0 \in \partial f(x^*)$$

KKT conditions revisited

Consider

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \ \forall i = 1, ..., m \end{aligned}$$

where $f, g_1, ..., g_m$ are convex but not differentiable everywhere.

Proposition

If strong duality holds, then necessary and sufficient conditions for primal and dual optimality of (λ^*, x^*) are

$$\begin{aligned} g_i(x^*) \leq 0, \quad & \forall i = 1, ..., m && \textit{(primal feasability)} \\ \lambda_i^* \geq 0, \quad & \forall i = 1, ..., m && \textit{(dual feasability)} \\ \lambda_i^* g_i(x^*) = 0, \quad & \forall i = 1, ..., m && \textit{(compl. slackness)} \\ 0 \in \partial L(x^*, \lambda^*) = \partial f(x^*) + \sum_{i=1}^{m} \lambda_i^* \partial g_i(x^*), & && \textit{(first order condition)} \end{aligned}$$

Proof : Follows the same line as for the differentiable case.

Basics in optimization

II - Algorithm(s)

From theory to practice

Back to the unconstrained optimization problem

$$\min_x \; f(x)$$

If f is differentiable, then $\forall \alpha, d, \; ||d||_2 = 1$:

$$\begin{aligned} f(x+\alpha d) &= f(x) + \alpha \nabla f(x)^T d + o(\alpha) \\ \Rightarrow \left| f(x+\alpha d) - f(x) \right| &\approx \alpha \left| \nabla f(x)^T d \right| \\ &\leq \alpha \left|\left| \nabla f(x) \right|\right|_2 \end{aligned}$$

Best direction : $-\dfrac{\nabla f(x)}{\left|\left|\nabla f(x)\right|\right|_2}$!

Gradient descent algorithm

Iterative procedure

$$\text{for } t = 1, \dots, T \qquad x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)})$$

$\alpha_t > 0$: step size parameter

Main difficulty : choice of $(\alpha_t)_t$.
⋆ constant stepsize,
⋆ decreasing stepsize,
⋆ "best" stepsize (a.k.a. steepest descent).

Both the convergence rate *and* the complexity depend on $(\alpha_t)_t$.

Example : Steepest gradient descent

Algorithm

Input $x_0,\ \varepsilon$

while $\left\|\nabla f(x^{(t)})\right\| \geq \varepsilon,$

$$x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)})$$

where $\alpha_t = \arg\min_{\alpha>0} \; f\left(x^{(t)} - \alpha_t \nabla f(x^{(t)})\right) \, (1)$

end

Properties

(i) $f(x^{(t+1)}) \leq f(x^{(t)})$ (descent property),
(ii) $< \nabla f(x^{(t+1)}), \nabla f(x^{(t)}) >$ (orthogonal directions),
(iii) If f is C^1 and strictly convex, then $(x^{(t)})$ converges to x^*.
Proof of (iii) : Technical ! See [CZ13].

Limitations

- ⋆ Solving (1) may be non-trivial
- ⋆ May be slow (see Accelerations, e.g. [N+07])

Alternative formulation of the gradient descent

Initial formulation

$$\text{At step } t+1, \qquad x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)})$$

Recasted as

$$x^{(t+1)} = \arg\min_x \left\{ f(x^{(t)}) + < \nabla f(x^{(t)}), x - x^{(t)} > + \frac{1}{2\alpha_t} \left\| x - x^{(t)} \right\|_2^2 \right\}$$

Interpretation

⋆ $f(x^{(t)}) + < \nabla f(x^{(t)}), x - x^{(t)} >$: linearization of f around $x^{(t)}$,

⋆ $\left\| x - x^{(t)} \right\|_2^2$: requires $x^{(t+1)}$ to be "not to far" from $x^{(t)}$,

⋆ α_t : rules the tradeoff.

Proximal gradient descent

$$\min_x \; f(x) + h(x)$$

f convex and differentiable (e.g. L_2 loss),
h convex but non differentiable (e.g. L_1 norm).

Linearize the differentiable part to obtain :

$$x^{(t+1)} = \arg\min_x \; \left\{ f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x - x^{(t)} \right\rangle + h(x) + \frac{1}{2\alpha_t} \left\| x - x^{(t)} \right\|_2^2 \right\}$$

Proximal operator

$$\mathrm{prox}_h(\theta) = \arg\min_z \; \left\{ \frac{1}{2} ||\theta - z||_2^2 + h(z) \right\}$$

In practice

1/ Compute the classical gradient step $x^{(t)} - \alpha_t \nabla f(x^{(t)})$,
2/ project according to the proximal operator

$$x^{(t+1)} = \mathrm{prox}_{\alpha_t h}\left(x^{(t)} - \alpha_t \nabla f(x^{(t)}) \right)$$

Application I : projected gradient descent

If minimization is subject to constraint $x \in \Omega \subsetneq \mathbb{R}^p$:

$$x^{(t+1)} = \arg\min_{x \in \Omega} \left\{ f(x^{(t)}) + < \nabla f(x^{(t)}), x - x^{(t)} > + \frac{1}{2\alpha_t} \left\| x - x^{(t)} \right\|_2^2 \right\}$$

$$= \arg\min_{x} \left\{ f(x^{(t)}) + < \nabla f(x^{(t)}), x - x^{(t)} > + \frac{1}{2\alpha_t} \left\| x - x^{(t)} \right\|_2^2 + I_\Omega(x) \right\}$$

where $I_\Omega(x) = \begin{cases} 0 \text{ if } x \in \Omega, \\ +\infty \text{ otherwise.} \end{cases}$

In practice

1/ Compute the classical gradient step $x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)})$,

2/ Project on Ω

$$x_{pr}^{(t+1)} = \Pi_\Omega \left(x^{(t+1)} \right).$$

Fast if projection can be easily computed...

Application II : projected gradient descent for lasso regression

$$x^{(t+1)} = \arg\min_{x} \left\{ f(x^{(t)}) + < \nabla f(x^{(t)}), x - x^{(t)} > + \frac{1}{2\alpha_t} \left\| x - x^{(t)} \right\|_2^2 + \lambda \|x\|_1 \right\}$$

In practice

1/ Compute the classical gradient step $x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)})$,

2/ Apply soft-thresholding to $x^{(t+1)}$

$$x_{pr,j}^{(t+1)} = \text{sign}\left[x^{(t+1)_j}\right] \times \left| \left| x^{(t+1)_j} \right| - \alpha_t \lambda \right|_+ .$$

Fast, easy, and amenable to parallelization.

Beyond first order algorithms

$$\min_x \quad f(x)$$

f convex and twice differentiable

Newton algorithm

⋆ Consider 2[nd] order Taylor expansion of f :

$$\begin{aligned} f(y) &= f(x) + < \nabla f(x), y - x > + \frac{1}{2}(y-x)^T H_f(x)(y-x) + o\left(||y-x||_2^2\right) \\ &= Q_x(y) + o\left(||y-x||_2^2\right) \end{aligned}$$

⋆ At step $t+1$, use $Q_{x^{(t)}}$ as a proxy for f...

$$x^{(t+1)} = \arg\min_x \quad Q_{x^{(t)}}(x)$$

⋆ ... and get the (closed form) solution :

$$x^{(t+1)} = x^{(t)} - H_f(x^{(t)})^{-1} \nabla f(x^{(t)})$$

Take home message

Theoretical aspects

- ⋆ Mostly interested in convex problems,
- ⋆ Characterization of the solution(s),
- ⋆ Guidelines to derive efficient algorithms.

Gradient descent

- ⋆ Simple but quite versatile,
- ⋆ Can be generalized in many ways,
- ⋆ More suited to deal with large problems than Newton method (more on this latter).

Non-addressed points

- ⋆ Complexity of the different algorithms
- ⋆ Rates of convergence
- ⋆ Convexity vs strong convexity, smoothness, etc.

Basics in classification

Elements of Statistical Learning [FHT01]
A Probabilistic Theory of Pattern Recognition [DGL13]

Supervised classification

Goal

Predict the unknown label Y of an observation X.

- $Y \in \mathscr{Y}$ where $\mathscr{Y} = \{0,1\}$ or $\mathscr{Y} = \{-1,1\}$ (binary classif.),
- $X \in \mathscr{X} (= \mathbb{R}^p)$.

Supervision

$\mathbb{P}_{X,Y}$ is unknown.

Training set : $\mathscr{D}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$, where $(X_i, Y_i) \overset{i.i.d.}{\hookrightarrow} \mathbb{P}_{X,Y}$.

Classifier

One aims at building

$$\begin{array}{rcl} \widehat{h} : \mathscr{X} & \rightarrow & \mathscr{Y} \\ X & \mapsto & \widehat{Y} \end{array}$$

Some examples

Cancer prediction

Predict cancer grade (from 1 to 3) based on CNV.

- ⋆ $X_i = (X_{i1}, ..., X_{ip})$, where
  X_{ij} = Nb of copies of chrom. segment j in ind. i.
- ⋆ $\mathscr{X} = \mathbb{R}^p$
- ⋆ $\mathscr{Y} = \{1, 2, 3\}$

Credit scoring

Predict loan reimbursement based on social/economics/health measurements.

- ⋆ $X_i = (X_{i1}, ..., X_{i3})$, where
  X_{i1} = gross salary of ind. i,
  $X_{i2} \in 1, ..., K$ = socio-professional category of ind. i,
  $X_{i3} = 1$ if ind. i already has an ongoing loan, 0 otherwise.
- ⋆ $\mathscr{X} = \mathbb{R} \times \{1, ..., K\} \times \{0, 1\}$
- ⋆ $\mathscr{Y} = \{\text{"safe"}, \text{"risky"}\}$

Pattern detection in images, Text categorization, ...

Classification algorithms

Any strategy

$$\begin{aligned}\mathscr{A} : \bigcup_{n\geq 1} \{\mathscr{X} \times \mathscr{Y}\}^n &\rightarrow \mathscr{Y}^{\mathscr{X}} \\ \mathscr{D}_n &\mapsto \widehat{h}\end{aligned}$$

defines a classification algorithm.

A few examples

- Discriminant analysis
- *k*NN
- Logistic regression
- Neural networks
- SVM
- CART & Random forest
- Boosting/bagging
- ...

Performance assessment

Quality of a classifier

$$L(\widehat{h}) = \mathbb{P}\left(\widehat{h}(X) \neq Y \ |\mathscr{D}_n\right) = \mathbb{E}\left[\ell_{HL}\left(Y, \widehat{h}(X)\right) \ |\mathscr{D}_n\right]$$

where
$$\begin{aligned} \ell_{HL}\left(Y, \widehat{h}(X)\right) &= I_{\left\{\widehat{h}(X) \neq Y\right\}} \quad (\text{case } \{0,1\}), \\ \ell_{HL}\left(Y, \widehat{h}(X)\right) &= I_{\left\{Y\widehat{h}(X) < 0\right\}} \quad (\text{case } \{-1,1\}). \end{aligned}$$

ℓ_{HL} : hard loss.

Empirical error rate

$$L_n(\widehat{h}) = \frac{1}{n} \sum_{i=1}^{n} \ell_{HL}\left(\widehat{h}(X_i), Y_i\right)$$

Bayes classifier

Assume - $\mathbb{P}_X$ has a density w.r.t. Lebesgue measure,
- $\eta(x) = \mathbb{P}(Y = 1 \mid X = x)$ is defined everywhere,

and define

$$h_B(x) = \begin{cases} 1 & \text{if } \eta(x) > 0.5 \\ 0 & \text{if } \eta(x) < 0.5 \\ \mathscr{B}(0.5) & \text{otherwise.} \end{cases}$$

Proposition

$$h_B = \arg\min_h L(h)$$

Some notations

In the following, we will consider classifiers of the form

$$h_f(x) = I_{\{f(x)>0\}} \quad \text{or} \quad h_f(x) = \text{sign}\,[f(x)]$$

Example 1 : Bayes classifier

$$h_B(x) = I_{\left\{\eta(x)-\frac{1}{2}>0\right\}} \quad \text{or} \quad h_B(x) = \text{sign}\left[\eta(x) - \frac{1}{2}\right]$$

Example 2 : linear classifier

$$h_\beta(x) = I_{\{x^T\beta>0\}} \quad \text{or} \quad h_\beta(x) = \text{sign}\left[x^T\beta\right]$$

Overview

Logistic regression

Statistical learning with sparsity [HTW15]

From LM to GLM

Linear (regression) model

$$Y_i = x_i\beta + \varepsilon_i,\ \varepsilon_i \hookrightarrow \mathcal{N}\left(0, \sigma^2\right),\ i.i.d. \quad \Leftrightarrow \quad Y_i|X_i = x_i \hookrightarrow \mathcal{N}\left(x_i\beta, \sigma^2\right),\ ind.$$

$$\Leftrightarrow \quad \begin{cases} Y_i|X_i = x_i \hookrightarrow \mathcal{N}\left(\mu_{x_i}, \sigma^2\right) \\ \quad \mu_{x_i} = x_i^T\beta \end{cases}$$

Generalized linear model

$$\begin{cases} Y_i|X_i = x_i \hookrightarrow \mathcal{B}(p_{x_i}),\ ind. \\ \quad p_{x_i} = g^{-1}\left(x_i{}^T\beta\right) \end{cases}$$

where $g(t) = \log\left[\frac{t}{1-t}\right]$ is the "logit" *link function*.

Note : Only $Y|x$ is considered.

Maximum likelihood inference

$Y_1, \ldots, Y_n$ independent cond. to $x_1, \ldots, x_n$,
$Y_i | x_i \hookrightarrow \mathscr{B}(p_{x_i})$, $\forall i = 1, \ldots, n$

$$\Rightarrow \mathscr{L}(\beta) = \log \left\{ \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i} \right\}$$

Proposition

$$\begin{aligned} \nabla \mathscr{L}(\beta) &= X^T (y - p), \\ H \mathscr{L}(\beta) &= -X^T D X, \end{aligned}$$

where $p = (p_1, \ldots, p_n)$, $D = diag(p_i(1 - p_i))$.

Note : No closed form solution for $\widehat{\beta}$ but $\mathscr{L}(\beta)$ is concave.
$\Rightarrow$ Numeric optimization via Newton algorithm.

Newton method for LR

Main steps

⋆ 2nd order approximation

$$\widetilde{\mathscr{L}}_{(t)}(\beta) = \mathscr{L}(\widehat{\beta}^{(t)}) + \left[\nabla \mathscr{L}(\widehat{\beta}^{(t)})\right]^T \left(\beta - \widehat{\beta}^{(t)}\right) + \frac{1}{2}\left(\beta - \widehat{\beta}^{(t)}\right)^T [H\mathscr{L}(\beta)] \left(\beta - \widehat{\beta}^{(t)}\right)$$

⋆ Define

$$\widehat{\beta}^{(t+1)} = \arg\max_{\beta} \ \widetilde{\mathscr{L}}_{(t)}(\beta)$$

Proposition

i) $\widehat{\beta}^{(t+1)} = \widehat{\beta}^{(t)} + \left[X^T D_{(t)} X\right]^{-1} X^T (y - p_{(t)})$,
ii) $\widehat{\beta}^{(t+1)}$ *is also solution of*

$$\arg\min_{\beta} \ ||X\beta - z_{(t)}||^2_{D^{-1}_{(t)}} \ ,$$

where $z_{(t)} = X\widehat{\beta}^{(t)} + D^{-1}_{(t)}(y - p_{(t)})$ *and* $p_{(t)} = \left(p_i(\widehat{\beta}^{(t)})\right)_{1 \le i \le n}$.

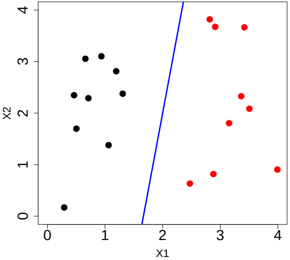
Logistic regression classifier

Proposition

The LR classifier is a linear classifier defined as

$$\widehat{h}_{LR}(x) = I_{\{x^T\widehat{\beta}>0\}} \quad \textit{where} \quad \widehat{\beta} = \arg\max_{\beta} \mathscr{L}(\beta)$$

Separability : definition



Definition

A training set is separable if there exists β *such that*

$$\forall i / y_i = 1,\ x_i^T \beta > 0$$
$$\forall i / y_i = 0,\ x_i^T \beta < 0$$

Note 1 : $\Leftrightarrow$ there exists a linear classifier h such that $L_n(h) = 0$,
Note 2 : discrete case : can be relaxed to a single cell.

Separability : consequence

Proposition

If the training set is separable, then

$$\begin{aligned} \mathscr{L}(\widehat{\beta}) &= 0, \\ \textit{and } ||\widehat{\beta}|| &= +\infty. \end{aligned}$$

⇒ Even in the "small dimension" setting, regularization may be required.

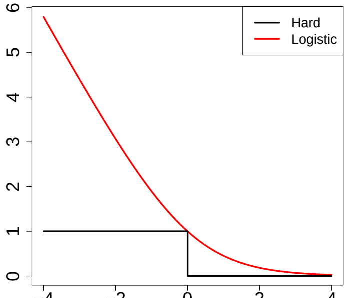
From MLE to convex risk minimization

Proposition

Assume $Y_i = \pm 1, \forall i$. One has

$$\begin{aligned} \widehat{h}_{LR}(x) &= sign\left[x^T\widehat{\beta}\right], \\ with \quad \widehat{\beta} &= \arg\min_{\beta} \sum_{i=1}^{n} \ell_{LR}\left(y_i x_i^T \beta\right) \end{aligned}$$

where $\ell_{LR}(t) = \log\left[1 + e^{-t}\right]$ is the logistic loss.



Regularized logistic regression

Definition

For any $\lambda > 0$ the regularized LR classifier is defined as

$$
\begin{aligned}
\widehat{h}^{\lambda}_{RLR}(x) &= sign\left[x^T\widehat{\beta}_\lambda\right], \\
\text{with} \quad \widehat{\beta}_\lambda &= \arg\min_{\beta} \sum_{i=1}^{n} \ell_{LR}\left(y_i x_i^T \beta\right) + \lambda R(\beta)
\end{aligned}
$$

Ridge LR : $R(\beta) = ||\beta||_2^2$
$\widehat{\beta}_\lambda$ is always defined and unique.

Lasso LR : $R(\beta) = ||\beta||_1$
$\widehat{\beta}_\lambda$ is always defined and unique (under mild conditions, [T⁺13]).

Inference for regularized logistic regression

Recall in the low dimensional case

$$\widehat{\beta}^{(t+1)} = \arg\min_{\beta} \ ||X\beta - z_{(t)}||^2_{D^{-1}_{(t)}} \ ,$$

where $z_{(t)} = X\widehat{\beta}^{(t)} + D^{-1}_{(t)}(y - p_{(t)})$ and $p_{(t)} = \left(p_i(\widehat{\beta}^{(t)})\right)_i$.

Solving regularized LR...
... is replaced with solving

$$\widehat{\beta}^{(t+1)}_{\lambda} = \arg\min_{\beta} \ \left\{||X\beta - z||^2_{D^{-1}_{(t)}} + \lambda R(\beta)\right\} \qquad (1)$$

hence boils down to regularized regression (at each step) !

Ridge LR : Solution of (1) has a closed form expression.
Lasso LR : use proximal/coordinate gradient descent.

Exact optimization for Lasso LR [SK03]

Solve

$$\widehat{\beta}, \widehat{\beta}_0 = \arg\min_{\beta, \beta_0} \left\{ \sum_{i=1}^{n} \ell_{LR}(y_i f(x_i)) + \lambda \|\beta\|_1 \right\}, \quad \text{where } f(x) = x^T \beta + \beta_0$$

First order conditions

Lead to the definition of the violation criterion :

$$\text{At point } \beta, \quad V_j = \begin{cases} |\lambda - F_j| & \text{if } \beta_j > 0 \\ |\lambda + F_j| & \text{if } \beta_j < 0 \\ \max(F_j - \lambda,\ -F_j - \lambda,\ 0) & \text{if } \beta_j = 0 \end{cases}$$

where

$$F_j = \sum_{i=1}^{n} \frac{e^{-y_i f(x_i)}}{1 + e^{-y_i f(x_i)}} y_i x_{ij}$$

Note : At point $\widehat{\beta}$, $V_j = 0 \ \forall j$.

Solving for a single value of λ

Require: $(x_1, y_1), ..., (x_n, y_n)$, λ

Initialize β to β_{init} ; Set $\mathscr{A} = \{j / \beta_j \neq 0\}$

while There exists $j \notin \mathscr{A}$ s.t. $V_j \neq 0$ **do**

  Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$

  Update $\mathscr{A} \leftarrow \mathscr{A} \cup \{j_{\max}\}$

  while there exists $j \in \mathscr{A}$ s.t. $V_j \neq 0$ **do**

    Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$

    Optimize $L(\beta)$ w.r.t. $\beta_{j_{\max}}$

    Recompute V_j, $j \in \mathscr{A}$

  end while

end while

return β

Solving for a single value of λ

Require: $(x_1, y_1), \ldots, (x_n, y_n)$, λ
  Initialize β to β_{init} ; Set $\mathscr{A} = \{j / \beta_j \neq 0\}$
  while There exists $j \notin \mathscr{A}$ s.t. $V_j \neq 0$ **do**
    Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$
    Update $\mathscr{A} \leftarrow \mathscr{A} \cup \{j_{\max}\}$
    while there exists $j \in \mathscr{A}$ s.t. $V_j \neq 0$ **do**
      Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$
      Optimize $L(\beta)$ w.r.t. $\beta_{j_{\max}}$
      Recompute V_j, $j \in \mathscr{A}$
    end while
  end while
  return β

Solving for a single value of λ

Require: $(x_1, y_1), \ldots, (x_n, y_n)$, λ
  Initialize β to β_{init} ; Set $\mathscr{A} = \{j / \beta_j \neq 0\}$
  while There exists $j \notin \mathscr{A}$ s.t. $V_j \neq 0$ **do**
    Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$
    Update $\mathscr{A} \leftarrow \mathscr{A} \cup \{j_{\max}\}$
    while there exists $j \in \mathscr{A}$ s.t. $V_j \neq 0$ **do**
      Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$
      Optimize $L(\beta)$ w.r.t. $\beta_{j_{\max}}$
      Recompute V_j, $j \in \mathscr{A}$
    end while
  end while
  return β

Solving for a single value of λ

Require: $(x_1, y_1), \ldots, (x_n, y_n)$, λ

Initialize β to β_{init} ; Set $\mathscr{A} = \{j / \beta_j \neq 0\}$

while There exists $j \notin \mathscr{A}$ s.t. $V_j \neq 0$ **do**

Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$

Update $\mathscr{A} \leftarrow \mathscr{A} \cup \{j_{\max}\}$

while there exists $j \in \mathscr{A}$ s.t. $V_j \neq 0$ **do**

Find $j_{\max} = \arg\max_{j \in \mathscr{A}} V_j$

Optimize $L(\beta)$ w.r.t. $\beta_{j_{\max}}$

Recompute V_j, $j \in \mathscr{A}$

end while

end while

return β

- ⋆ Sub-problem is str. convex $\Rightarrow$ $L(\beta)$ decreases at each step,
- ⋆ Only a sparse vector to store.
- ⋆ $\beta_{init} = 0$ seems perfect.

Solving for a set of λ values

Require: $(x_1, y_1), ..., (x_n, y_n)$, $\lambda_1 > ... > \lambda_m$
for k=1,...,m **do**
Initialize β^{λ_k} to $\widehat{\beta}^{\lambda_{k-1}}$; Set $\mathscr{A} = \left\{ j / \ \beta_j^{\lambda_k} \neq 0 \right\}$
while There exists $j \notin \mathscr{A}$ s.t. $V_j \neq 0$ **do**
Find $j_{\max} = \arg\max_{j \in \mathscr{A}} \ V_j$
Update $\mathscr{A} \leftarrow \mathscr{A} \cup \{ j_{\max} \}$
while there exists $j \in \mathscr{A}$ s.t. $V_j \neq 0$ **do**
Find $j_{\max} = \arg\max_{j \in \mathscr{A}} \ V_j$
Optimize $L(\beta^{\lambda_k})$ w.r.t. $\beta^{\lambda_k}_{j_{\max}}$
Recompute V_j, $j \in \mathscr{A}$
end while
end while
end for
return $\beta^{\lambda_1}, ..., \beta^{\lambda_m}$

Take home message

Logistic regression

⋆ Belongs to the GLM family,

⋆ Is a linear classifier,

⋆ Is also an ECRM minimizer.

⋆ May require regularization even in small dimension

Inference

⋆ Can be performed easily,

⋆ But cannot be performed easily !

⇒ Pay attention to which package you use...

Overview

Support Vector Machines

Learning With Kernels [SS01]
Kernels Methods In Computational Biology [STV04]

Back to basics

Bayes classifier

$$h_B = \arg\min_h \; L(h)$$

- Requires $\mathbb{P}_{X,Y}$,
- $\mathscr{Y}^{\mathscr{X}}$ is... large !

Find a linear classifier

$$\widehat{h}_{\widehat{f}} = \arg\min_{h_f} \; L_n(h_f), \quad \text{where } f(x) = x^T\beta + \beta_0$$

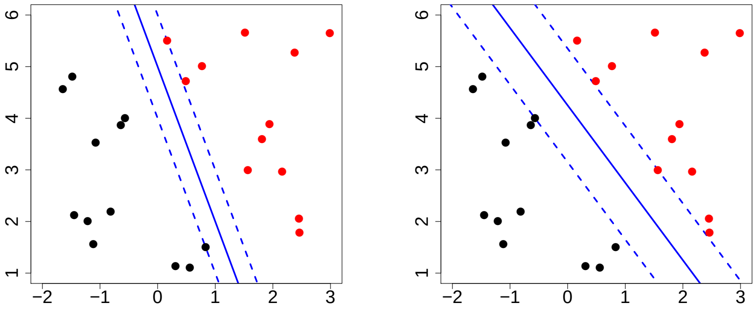
- Not unique (in two ways),
- Still NP hard to find in practice...

Find the *optimal* linear classifier

$$\widehat{h}_{\widehat{f}} = \text{sign}\left[\widehat{f}(x)\right] \quad \text{where } \widehat{f}(x) = x^T\widehat{\beta} + \widehat{\beta}_0 \text{ and } \widehat{\beta}, \widehat{\beta}_0 = \arg\min_{\beta,\beta_0} \; Crit(\beta,\beta_0)$$

Separating hyperplanes and margin

Consider a linearly separable dataset



Separating hyperplane : Any $\Delta : \{x^T\beta + \beta_0 = 0\}$ s.t.

$$y_i\left(x_i^T\beta + \beta_0\right) > 0.$$

Margin Smallest distance between a point and Δ.

Maximum margin hyperplane

If the dataset is linearly separable, choose $(\widehat{\beta}_0, \widehat{\beta})$ such that

$$\widehat{\Delta} = \left\{ x^T \widehat{\beta} + \widehat{\beta}_0 \right\} = 0$$

has **maximum** margin.

Proposition

$$(\widehat{\beta}_0, \widehat{\beta}) = \arg\min_{\beta_0, \beta} \frac{1}{2} ||\beta||_2^2 \ u.c. \quad y_i \left(x_i^T \beta + \beta_0 \right) > 1, \ \forall i$$

Note : Constraints account for - correct classification
- maximum margin
- Δ representation identification.

Non separable case

If the dataset is not linearly separable, relax constraints as follows

$$y_i\left(x_i^T\beta+\beta_0\right)\geq 1-\varepsilon_i,\ \forall i$$

where $\varepsilon_i \geq 0$, and penalize for the extend of margin violation.

Definition (Soft Margin SVM classifier)

$$\widehat{h}_{SVM}=sign\left[\widehat{\beta}_0+x^T\widehat{\beta}\right]$$

where $(\widehat{\beta}_0,\widehat{\beta})$ *is solution of*

$$\begin{aligned}\textit{minimize}\quad & \frac{1}{2}\|\beta\|_2^2+C\sum_{i=1}^{n}\varepsilon_i\\ \textit{subject to}\quad & y_i\left(x_i^T\beta+\beta_0\right)\geq 1-\varepsilon_i,\ \forall i=1,\dots,n\\ & \varepsilon_i\geq 0,\ \forall i=1,\dots,n\end{aligned}$$

Inference

Inference boils down to solving the following problem :

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\beta||_2^2 + C\sum_{i=1}^{n} \varepsilon_i \\ \text{subject to} \quad & y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0 \end{aligned}$$

Associated primal problem :

$$\min_{\beta,\beta_0,\varepsilon} \; \max_{\alpha,\mu\geq 0} \; L(\beta,\beta_0,\varepsilon,\alpha,\mu)$$

where

$$L(\beta,\beta_0,\varepsilon,\alpha,\mu) = \frac{1}{2}||\beta||_2^2 + C\sum_{i=1}^{n}\varepsilon_i + \sum_{i=1}^{n}\alpha_i\left[1 - \varepsilon_i - y_i\left(x_i^T\beta + \beta_0\right)\right] - \sum_{i=1}^{n}\mu_i\varepsilon_i$$

Dual optimization problem

Dual problem :

$$\max_{\alpha,\mu \succeq 0} \; \min_{\beta,\beta_0,\varepsilon} \; L(\beta,\beta_0,\varepsilon,\alpha,\mu)$$

Proposition

The SVM dual problem can be reformulated as

$$\begin{aligned} \textit{maximize} \quad & -\frac{1}{2}\alpha^T Q\alpha + 1_n^T\alpha \\ \textit{subject to} \quad & 0 \preceq \alpha \preceq C \\ & \textstyle\sum_{i=1}^{n} y_i\alpha_i = 0 \end{aligned}$$

where $Q_{ij} = y_i y_j x_i^T x_j$.

Sequential Minimal Optimization [Pla98]

Let B a subset of $\{1,...,n\}$. One has

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\begin{bmatrix}\alpha_B\\ \alpha_{\overline{B}}\end{bmatrix}\begin{bmatrix}Q_{BB} & Q_{B\overline{B}}\\ Q_{\overline{B}B} & Q_{\overline{B}\overline{B}}\end{bmatrix}\begin{bmatrix}\alpha_B\\ \alpha_{\overline{B}}\end{bmatrix} + 1_{|B|}^T\alpha_B + 1_{|\overline{B}|}^T\alpha_{\overline{B}} \\ \text{subject to} & 0 \preceq \alpha_B \preceq C,\ 0 \preceq \alpha_{\overline{B}} \preceq C \\ & Y_B^T\alpha_B + Y_{\overline{B}}^T\alpha_{\overline{B}} = 0 \end{array}$$

$\Leftrightarrow$

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\alpha_B^T Q_{BB}\alpha_B + U(\overline{B})^T\alpha_B + \Delta_1(\overline{B}) \\ \text{subject to} & 0 \preceq \alpha_B \preceq C \\ & Y_B^T\alpha_B = \Delta_2(\overline{B}) \end{array}$$

Apply with $|B| = 2$!

⋆ Simpler optimization problem,

⋆ Only 2 columns of Q need to be loaded at each step,

⋆ "Pairwise" coordinate descent.

Note : One can search for the "best" pair at each step...

From SVM to convex risk minimization

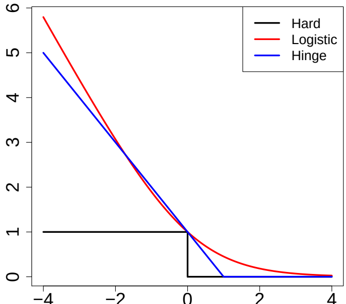
Proposition

Assume $Y_i = \pm 1, \forall i$. One has

$$\widehat{h}^{\lambda}_{SVM}(x) = sign\left[x^T\widehat{\beta}_\lambda + \widehat{\beta}_{0\lambda}\right],$$

$$\text{with} \quad (\widehat{\beta}_{0\lambda}, \widehat{\beta}_\lambda) = \arg\min_{\beta_0,\beta} \sum_{i=1}^{n} \ell_{SVM}\left(y_i x_i^T \beta\right) + \lambda \|\beta\|_2^2$$

where $\ell_{SVM}(t) = |1-t|_+$ is the hinge loss.



So far...

SVM classifier

$$
\begin{aligned}
\widehat{h}^{\lambda}_{SVM}(x) &= \text{sign}\left[x^T\widehat{\beta}_\lambda + \widehat{\beta}_{0\lambda}\right], \\
\text{with} \quad (\widehat{\beta}_{0\lambda}, \widehat{\beta}_\lambda) &= \arg\min_{\beta_0,\beta} \sum_{i=1}^{n} \ell_{SVM}\left(y_i x_i^T \beta\right) + \lambda \|\beta\|_2^2
\end{aligned}
$$

⋆ Linear classifier with largest margin,
⋆ Linear classifier that minimizes the hinge loss.

Inference

$$
\begin{aligned}
\widehat{\beta}_\lambda &= \sum_{i=1}^{n} y_i \widehat{\alpha}_i x_i \\
\text{with} \quad \widehat{\alpha} &= \arg\max_{0\preceq\alpha\preceq 1/\lambda} \left\{ -\frac{1}{2}\alpha^T Q \alpha + 1_n^T \alpha \right\} \quad \text{u.c.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0
\end{aligned}
$$

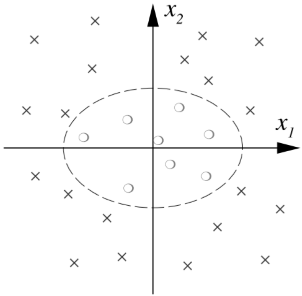
So far...

Restricted to $\mathscr{X} = \mathbb{R}^p$.

What about - text classification ?

- sequence classification ?
- pathway classification ?
- ...

Restricted to *linear* classification :



Naive way : transform and proceed (1/3)

Example 1 Document classification (e.g. Reuters dataset)

Bag of words

⋆ $\mathscr{Y} = \{1, ..., M\}$, with M the number of document classes,

⋆ Apply transformation $\phi : \mathscr{X} = \{\text{documents}\} \to \mathbb{R}^d$

$$\phi(doc) = (N_{w_1}, ..., N_{w_d}),$$

where N_{w_j} is the nb. of occurrence of word w_j in *doc*.

Characteristics

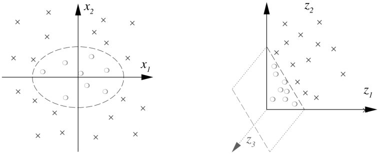
⋆ d is large ($\approx$ 35k),

⋆ $\phi(doc)$ is sparse (between 93 and 1263 words per doc).

Storing the $\phi(doc)$'s is cheap !

Naive way : transform and proceed (2/3)

Example 2 Non-linear classification (e.g. Sphere example)



Apply transformation

$$x = (x_1, x_2)^T \mapsto \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2).$$

Characteristics

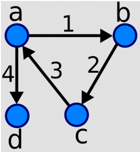
⋆ $x \in \mathbb{R}^p$ is "big", $\phi(x)$ is way bigger,
⋆ $\phi(x)$ is not sparse.

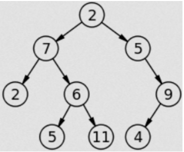
Storing the $\phi(x)$'s is prohibitive.

Naive way : transform and proceed (3/3)

Example 3 Structured data classification

Networks, trees





Sequences

```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
   ||||||||||    |||||||  ||||||||||||| |||||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
```

Finding $\phi : \mathscr{S} \rightarrow \mathbb{R}^d$ is non-trivial.

Smart way : kernel SVM

Combining CRM formulation + Inference leads to :

$$\widehat{h}_{SVM}(x) = \text{sign}\left[\sum_{i=1}^{n} y_i \widehat{\alpha}_i \langle x_i, x\rangle + c_{\widehat{\alpha}}\right],$$

$$\text{with} \quad \widehat{\alpha} = \arg\max_{0 \leq \alpha \leq 1/\lambda} \left\{\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} y_i y_{i'} \alpha_i \alpha_{i'} \langle x_i, x_{i'}\rangle\right\}$$

$$\text{u.c.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

The x_i's only appear through scalar products.

$\Rightarrow$ Only need to compute $< \phi(x_i), \phi(x_j) >$.

$\Rightarrow$ Only need to store the $n \times n$ Gram matrix.

Smart way : kernel SVM

Combining CRM formulation + Inference leads to :

$$\hat{h}_{SVM}(x) = \text{sign}\left[\sum_{i=1}^{n} y_i \hat{\alpha}_i k(x_i, x) + c_{\hat{\alpha}}\right],$$

$$\text{with} \quad \hat{\alpha} = \arg\max_{0 \preceq \alpha \preceq 1/\lambda} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} y_i y_{i'} \alpha_i \alpha_{i'} k(x_i, x_{i'}) \right\}$$

$$\text{u.c.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

The x_i's only appear through scalar products.

⇒ Only need to compute $< \phi(x_i), \phi(x_j) >$.

⇒ Only need to store the $n \times n$ Gram matrix.

⇒ Only need to compute some similarity between x_i and x_j.

Kernels is what we need !

- Stephen Boyd and Lieven Vandenberghe.
  Convex optimization.
  Cambridge university press, 2004.

- Edwin KP Chong and Stanislaw H Zak.
  An introduction to optimization, volume 76.
  John Wiley & Sons, 2013.

- Luc Devroye, László Györfi, and Gábor Lugosi.
  A probabilistic theory of pattern recognition, volume 31.
  Springer Science & Business Media, 2013.

- Jerome Friedman, Trevor Hastie, and Robert Tibshirani.
  The elements of statistical learning, volume 1.
  Springer series in statistics New York, 2001.

- Christophe Giraud.
  Introduction to high-dimensional statistics, volume 138.
  CRC Press, 2014.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright.
Statistical learning with sparsity : the lasso and generalizations.
CRC press, 2015.

Yurii Nesterov et al.
Gradient methods for minimizing composite objective function, 2007.

John Platt.
Sequential minimal optimization : A fast algorithm for training support vector machines.
1998.

Shirish Krishnaj Shevade and S Sathiya Keerthi.
A simple and efficient algorithm for gene selection using sparse logistic regression.
Bioinformatics, 19(17) :2246–2253, 2003.

Bernhard Scholkopf and Alexander J Smola.
Learning with kernels : support vector machines, regularization, optimization, and beyond.
MIT press, 2001.

Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert.
Kernel methods in computational biology.
MIT press, 2004.

Ryan J Tibshirani et al.
The lasso problem and uniqueness.
Electronic Journal of Statistics, 7 :1456–1490, 2013.