Machine Learning And Applications: Supervised Learning-SVM

Raphaël Bournhonesque

École Normale Supérieure de Lyon, Lyon, France
raphael.bournhonesque@ens-lyon.fr

1 Supervised vs unsupervised learning

Machine learning algorithms can be either supervised or unsupervised. Supervised learning consists of using already labeled data (training set) to infere the labels of new data. The label can be a class (classification problem), a real value (regression problem).

Unsupervised learning is adapted when we do not have any information on the class of the data.

2 Penalized empirical risk minimization

During the last course, we saw a method to minimize the structural risk, by:

- defining nested function sets of increasing complexity
- minimizing the empirical risk over each family
- choosing the solution that gives the best generalization performances

This approach requires to solve a constrained optimization problem, which is usually harder than to solve its unconstrained equivalent.

This problem is often equivalent to the following penalized problem:

$$\min_f \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega(f)$$

The first term corresponds to the loss function minimization, and favors a good fit to the data. The second term is a penalty term, which penalizes functions with high complexity, it thus favors regularity of f. In order to solve this problem, we need to define a good mesure of the complexity Ω, and then to compare the generalization performances of the functions found for decreasing values of λ.

2.1 Loss functions

For regression problems, common loss functions include the l_1 or l_2 norms of $y - f(x)$. The l_1 norm has the advantage of being robust (i.e: less sensitive to large errors than the l_2 norm). With the l_2 norm, if a function makes one large mistake, it can be very costly.

For classification problems, we can use the 0/1, the logistic or the hinge loss function. The 0/1 loss function has a major drawback: it is a non-convex function, and optimization problems are simpler to solve when the function to minimize is convex. Indeed, some general results of convex optimization guarantee the existence of a unique global solution if a minimum is found.

Moreover, methods based on convex objectives are simpler to analyze.

2.2 Penalties

Let us consider a linear function $f(x) = \Theta^T x, \Theta \in \mathbb{R}^p$.

A very common penalty is the ridge penalty:

$$\Omega(\Theta) = \|\Theta\|_2^2$$

The ridge penalty is used in ridge regression (a regression algorithm, combined with the l_2 loss) and in Support Vector Machines (SVM) (a classification algorithm, combined with the hinge loss).

This penalty leads to function that are regular in the sense that 2 points x, x' that are closed in the Euclidian space have close evaluations by the function, since by the Cauchy-Schwarz inequality:

$$|\Theta^T x - \Theta^T x'| \leq \|\Theta\|_2 \|x - x'\|_2$$

This property can limit overfitting and improve generalization performances: it makes functions behave in a similar manner over similar data.

3 The ridge regression

We now assume that the data can be explained with a linear model:

$$y = \bar{\Theta}^T x + \epsilon$$

where ϵ is a random noise with mean 0 and variance σ^2, $y \in \mathbb{R}$ and $\Theta, x \in \mathbb{R}^p$. We observe n realization of this linear model, represented by the matrix $X \in \mathbb{R}^{n,p}$, and the vector $Y \in \mathbb{R}^n$.

Let us consider the estimator:

$$\hat{\Theta} = \arg\min_{\Theta}(\|Y - X\Theta\|^2 + \lambda\|\Theta\|^2)$$

We can show that:

$$\hat{\Theta} = (X^T X + \lambda I)^{-1} X^T Y$$

It is also possible the find the bias $E[\hat{\Theta} - \bar{\Theta}]$ and the variance of $\hat{\Theta}$:

$$E[\hat{\Theta} - \bar{\Theta}] = -\lambda(X^T X + \lambda I)^{-1}\bar{\Theta}$$

$$Var[\hat{\Theta}] = \sigma^2 (X^T X + \lambda I)^{-1} X^T X (X^T X + \lambda I)^{-1}$$

The bias increases with λ and tends to $-\bar{\Theta}$ when $\lambda \longrightarrow \infty$. It is noteworthy that if $\lambda = 0$ (unpenalized linear regression), the bias is null. If the ratio n/p is small (i.e the number of parameters is large compared to the number of points), the bias and the variance increase.

In practise, for a given dataset, some λ give smaller risks than others. λ can be chosen by hold-out or cross validation.

4 Fundamentals of constrained optimization

Constrained optimization methods are important to understand the SVM. Let us consider an equality and inequality constrained optimization problem over a variable $x \in \mathscr{X}$:
minimize $f(x)$
subject to $h_i(x) = 0,\ i = 1, \ldots, m,\ g_i(x) \leq 0,\ j = 1, \ldots, r,$ making no assumption about f, g and h.

Let f^* be the optimal value of the decision function under the constraints ($f^* = f(x^*)$ if the global minimum is reached at x^*). The Lagrangian of the constrained problem is the function $L : \mathscr{X} \times \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}$ defined by:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{r} \mu_j g_j(x)$$

We can now define the Lagrange dual function $q : \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}$:

$$q(\lambda, \mu) = \inf_{x \in \mathscr{X}} L(x, \lambda, \mu)$$

We can show that q is concave in (λ, μ), even if the original problem in not convex. Furthermore, we have:

$$q(\lambda, \mu) \leq f^*,\ \forall \lambda \in \mathbb{R}^m, \forall \mu \in \mathbb{R}^r, \mu \geq 0$$

We can finally define the Lagrange dual problem:
maximize $q(\lambda, \mu)$ **subject to** $\mu \geq 0$
If we let d^* be the optimal value of Lagrange dual problem, the **weak duality inequality** is:

$$d^* \leq f^*$$

The difference $d^* - f^*$ is called the **optimal duality gap**.
The **strong duality** holds when:

$$d^* = f^*$$

Strong duality does not hold for general nonlinear problems, but usually holds for convex problems. The conditions that ensure strong duality for convex problems are called **constraint qualification**.

5 Penalized vs constrained risk optimization

6 Support vector machines

Support Vector Machines (SVM) is a popular linear classification algorithm. Let us consider a training set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$. Is is assumed that the set of points is linearly separable, i.e. that there exists $(w, b) \in \mathbb{R}^p \times \mathbb{R}$ such that:

$$\begin{cases} w.x_i + b > 0 \; if \; y_i = 1 \\ w.x_i + b < 0 \; if \; y_i = -1 \end{cases}$$

Many decision boundaries can split the training data into the two classes correctly, but which one offers the best generalization performance? Among the possible boundaries, SVM select the one with the largest margin between the two classes. The margin is defined as the distance between a planar decision surface that separates two classes and the closest training samples to the decision surface.

We can show that the margin is $\frac{2}{\|w\|}$. The values of w and b are chosen such that the margin is maximal under the constraint that the points in the training set are correctly classified (i.e. the decision function gives the correct value of y_i). The decision function gives the correct value of y_i if:

$$y_i(w.x_i + b) \geq 1$$

Maximizing the margin is equivalent to minimizing $\|w\|^2$. Hence, finding the optimal hyperplace fullfilling these conditions can be written as a constrained optimization problem:

Find (w, b) which minimize $\|w\|^2$ under the constraints:

$$\forall i = 1, \ldots, n, y_i(w.x_i + b) - 1 \geq 0$$

.

Using the constrained optimization methods studied earlier, we can obtain the Lagrange dual function:

$$q(\alpha) = \inf_{w \in \mathbb{R}^p, b \in \mathbb{R}} L(w, b, \alpha)$$

Then $q(\alpha)$ equals:

$$\begin{cases} \sum_{i=1}^n \alpha_i - 1/2 \sum_{j=1}^n y_i y_j \alpha_i \alpha_j x_i x_j & \text{if } \sum_{i=1}^n \alpha_i y_i = 0 \\ -\infty & \text{otherwise} \end{cases}$$

And the problem becomes: **maximize** $q(\alpha)$ under the constraints $\alpha \geq 0$.

This is a quadratic program on $\mathbb{R}^n$, and can be solved efficiently using dedicated optimization software. Once the optimal α $(= \alpha^*)$ is found, we can recover the (w^*, b^*) corresponding to the optimal hyperplane that separate the two classes:

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i$$

And the decision function is therefore:

$$f^*(x) = w^* x + b^*$$

This SVM method is called hard-margin, because the data is assumed to be linearly separable. If it is not, the problem cannot be solved anymore. The separation constraints need to be relaxed, by introducing slack variables ξ_i (soft-margin SVM):

$$\forall i = 1, \ldots, n,\ y_i(w.x_i + b) \geq 1 - \xi_i$$