Linear classifiers and Support Vector Machines (SVM)

Sylvain Le Corff

1. Reminder on Logistic regression

2. Linear Support Vector Machine (SVM)

Summary

Classification

Setting

→ Historical data about individuals $i = 1, \ldots, n$.

→ **Features** vector $X_i \in \mathbb{R}^d$ for each individual i.

→ For each i, X_i belongs to a group ($Y_i = 0$) or not ($Y_i = 1$).

→ $Y_i \in \{0, 1\}$ is the **label** of i.

Objective

→ Given a new feature vector, predict a label in $\{0, 1\}$.

→ Use data $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ to construct a **classifier**.

Best Solution

The best solution f^* (which is independent of $\mathcal{D}_n$) is

$$f^* = \text{argmin}_{f:\mathbb{R}^d \to \{0,1\}} \mathbb{E}[\mathbb{1}_{Y \neq f(X)}] = \text{argmin}_{f:\mathbb{R}^d \to \{0,1\}} \mathbb{P}(Y \neq f(X))\,.$$

Bayes Predictor (explicit solution)
Binary classification with $0-1$ loss:

$$f^*(X) = \begin{cases} +1 & \text{if} \quad \mathbb{P}(Y=1|X) \geqslant \mathbb{P}(Y=0|X) \\ & \;\Leftrightarrow \mathbb{P}(Y=1|X) \geqslant 1/2\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

The explicit solution requires to know the conditional law of Y given X...

Conditional law of Y given X?

Fully parametric modeling.
Estimate the law of (X, Y) and use the **Bayes formula** to deduce an estimate of the conditional law of Y: *LDA/QDA, Naive Bayes...*

Parametric conditional modeling.
Estimate the conditional law of Y by a **parametric** law: *linear regression, logistic regression, Feed Forward Neural Networks...*

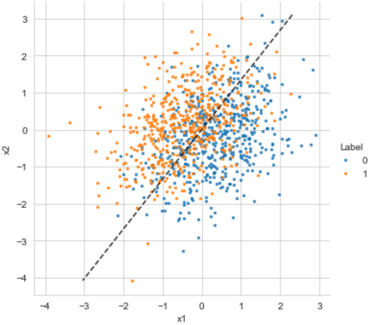
Nonparametric conditional modeling.
Estimate the conditional law of Y by a **non parametric** estimate: *kernel methods, nearest neighbors...*

Fully parametric modeling - Discriminant Analysis

In the LDA case, the classification rule is of the form:

$$f^*(x) = 1 \Leftrightarrow \langle w, x \rangle + b \geqslant 0\,,$$

where w and b depends on the model parameters.



→ Relax the Gaussian assumption ? (logistic model, SVM).
→ Design nonlinear classification rules ? (kernels, neural networks).

The logistic model

- One of the most widely used classification algorithm.
- Logistic model is generalized linear model of the linear model in the context of binary classification ($\mathcal{Y} = \{0, 1\}$).
- It models the distribution of Y given X. For $y \in \{0, 1\}$

$$\mathbb{P}(Y = 1|X) = \sigma\left(X^\top w + b\right)$$

where $\sigma : z \mapsto (1 + \mathrm{e}^{-z})^{-1}$, $w \in \mathbb{R}^d$ is a vector of model weights and $b \in \mathbb{R}$ is the intercept, and where σ is the sigmoid function:

Some comments on the logistic model

- The sigmoid is a modelling choice to map $\mathbb{R} \to [0, 1]$ (to model a probability).
- We could also consider

$$\mathbb{P}(Y = 1|X) = F\left(X^\top w + b\right)$$

for any distribution function F.
- Another popular choice is the Gaussian distribution

$$F(z) = \mathbb{P}(\mathcal{N}(0, 1) \leqslant z),$$

which leads to the probit model.

The logistic model

- In the case of the sigmoid (logistic regression),

$$\mathbb{P}(Y = 1|X) = \frac{\exp(b + w^\top X)}{1 + \exp(b + w^\top X)} = \frac{1}{1 + \exp(-(b + w^\top X))}$$

$$\mathbb{P}(Y = 0|X) = \frac{1}{1 + \exp(b + w^\top X)}$$

- Log-odd ratio:

$$\log\left(\frac{\mathbb{P}(Y = 1|X)}{\mathbb{P}(Y = 0|X)}\right) = X^\top w + b.$$

Logistic regression - likelihood function

Compute $\hat{w}_n$ and $\hat{b}_n$ as follows:

$$(\hat{w}_n, \hat{b}_n) \in \mathrm{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n}\left(-Y_i(X_i^\top w + b) + \log(1 + e^{X_i^\top w + b})\right) .$$

$\rightharpoondown$ It is an **average of losses**, one for each sample point.
$\rightharpoondown$ It is a convex and smooth problem.

Using the **logistic loss** function

$$\ell : (y, y') \mapsto \log(1 + e^{-yy'})$$

yields

$$(\hat{w}_n, \hat{b}_n) \in \mathrm{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, \langle w, X_i\rangle + b) .$$

Summary

Binary classification problem

- Training dataset of pairs (X_i, Y_i) for $1 \leqslant i \leqslant n$.
- Features $X_i \in \mathbb{R}^d$ and labels $Y_i \in \{-1, 1\}$.
- Given a features vector $x \in \mathbb{R}^d$, we want to predict its associated label.
- Focus on linear classification, i.e. classifiers defined by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

$$h(x) = \mathrm{sign}(x^\top w + b)\,.$$

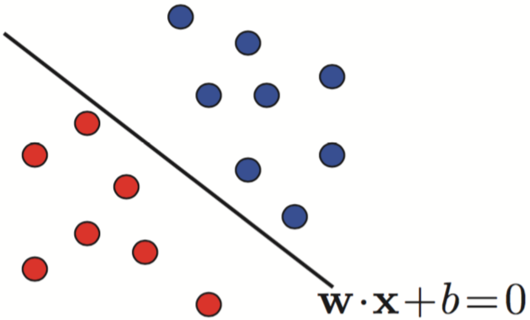
Preliminary definitions

A dataset is **linearly separable** if there exists an hyperplane H (linear classification rule) such that the following assumptions hold.

→ Points $X_i \in \mathbb{R}^d$ such that $Y_i = 1$ are on one side of the hyperplane.

→ Points $X_i \in \mathbb{R}^d$ such that $Y_i = -1$ are on the other side.

→ H does not pass through any point X_i.



Some geometry

A **hyperplane** is a translation of a set of vectors orthogonal to w.

$$H_{w,b} = \{x \in \mathbb{R}^d : w^\top x + b = 0\}\,.$$

$\rightharpoondown$ $w \in \mathbb{R}^d$ is a **non-zero vector normal** to the hyperplane.
$\rightharpoondown$ $b \in \mathbb{R}$ is a scalar.

Following for instance the results obtained for linear discriminant analysis and logistic regression, a hyperplane $H_{w,b}$ may be used as a classifier by defining

$$h_{w,b} : x \mapsto \begin{cases} 1 & \text{if } \langle w; x \rangle + b > 0\,, \\ -1 & \text{otherwise}\,. \end{cases}$$

Canonical hyperplane

If H do not pass through any sample point x_i, we can scale w and b so that

$$\min_{(x,y)\in D_n} |w^\top x + b| = 1$$

For such w and b, we call H the canonical hyperplane

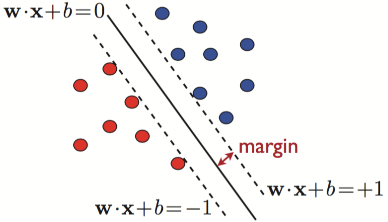


Figure: The marginal hyperplanes are the hyperplanes parallel to the separating hyperplane and passing through the closest points on the negative or positive sides.

The margin

The distance of any point $x \in \mathbb{R}^d$ to H is given by

$$d(x, H_{w,b}) = \frac{|\langle w, x \rangle + b|}{\|w\|}$$

So, if H is a canonical hyperplane, its margin is given by

$$\min_{(x,y) \in D_n} \frac{|w^\top x + b|}{\|w\|} = \frac{1}{\|w\|}.$$

Summary

If $\mathcal{D}_n$ is strictly linearly separable, we can find a canonical separating hyperplane

$$H_{w,b} = \{x \in \mathbb{R}^d : w^\top x + b = 0\},$$

that satisfies

$$|\langle w, X_i \rangle + b| \geqslant 1 \text{ for any } i = 1, \ldots, n,$$

which entails that a point X_i is correctly classified if

$$Y_i(\langle X_i, w \rangle + b) \geqslant 1.$$

The margin of H is equal to $1/\|w\|$.

Linear SVM: separable case

Hard Support Vector Machines is a classification procedure which aims at building a linear classifier with the largest possible margin, i.e. the largest minimal distance between a point in the training set and the hyperplane.

The hyperplane which **correctly separates all training data sets with the largest margin** is obtained with:

$$(\widehat{w}_n, \widehat{b}_n) \in \underset{\substack{(w,b)\in\mathbb{R}^d\times\mathbb{R};\|w\|=1, \\ \forall i\in\{1,\dots,n\},\, Y_i(\langle w;X_i\rangle+b)>0}}{\mathrm{argmax}} \left\{ \min_{1\leqslant i\leqslant n} |\langle w; X_i\rangle + b| \right\} .$$

The **hard Support Vector Machines** procedure is equivalent to solving the following optimization problem:

$$(\widehat{w}_n, \widehat{b}_n) \in \underset{(w,b)\in\mathbb{R}^d\times\mathbb{R}; \|w\|=1}{\operatorname{argmax}} \left\{ \min_{1\leqslant i\leqslant n} Y_i \left(\langle w; X_i\rangle + b\right) \right\} ,$$

A solution to the hard Support Vector Machines optimization problem is obtained by setting $(\widehat{w}_n, \widehat{b}_n) = (w_\star/\|w_\star\|, b_\star/\|w_\star\|)$ where

$$(w_\star, b_\star) \in \underset{\substack{(w,b)\in\mathbb{R}^d\times\mathbb{R}\\ \forall i\in\{1,\ldots,n\}, Y_i(\langle w;X_i\rangle+b)\geqslant 1}}{\operatorname{argmin}} \|w\|^2 .$$

Proof on blackboard !

Linear SVM: separable case

Maximum margin problem

In the Hard SVM case, a way of classifying $\mathcal{D}_n$ with maximum margin is to solve the following problem:

$$(w_\star, b_\star) \in \mathrm{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \left\{ \frac{1}{2} \|w\|_2^2 \right\} ,$$

under the constraints:

$$Y_i(\langle X_i, w \rangle + b) \geqslant 1 \, ,$$

for all $1 \leqslant i \leqslant n$.

- This problem admits a unique solution.
- It is a quadratic programming' problem.
- Dedicated optimization algorithms can solve this on a large scale very efficiently

Tools from constrained convex optimization

Consider a constrained optimization problem:

$$P^\star = \min_{x \in \mathbb{R}^d} \quad f(x)$$

under the constraints, for all $1 \leqslant i \leqslant p$, $1 \leqslant j \leqslant q$,

$$h_i(x) = 0 \quad \text{and} \quad g_j(x) \leqslant 0\,,$$

where $f, h_1, \ldots h_p, g_1, \ldots, g_q$ are defined on $\mathbb{R}^d$.

Lagrangian

The **Lagrangian** is the function defined on $\mathbb{R}^d \times \mathbb{R}^p \times \mathbb{R}^q_+$ by

$$\mathcal{L}(x, \lambda, \mu) :\mapsto f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x)$$

$\lambda \in \mathbb{R}^p$, $\mu \in \mathbb{R}^q_+$ are the Lagrange or dual variables.

The Lagrange dual function is defined by:

$$D : (\lambda, \mu) \mapsto \inf_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda, \mu) \,.$$

Let $\mathcal{D}$ be the subset of $\mathbb{R}^d$ of feasible points. Using

$$\sup_{\mu \geqslant 0, \lambda} \inf_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda, \mu) \leqslant \inf_{x \in \mathbb{R}^d} \sup_{\mu \geqslant 0, \lambda} \mathcal{L}(x, \lambda, \mu) = \inf_{x \in \mathrm{D}} f(x)$$

yealds the weak duality relation:

$$D^\star = \sup_{\mu \geqslant 0, \lambda} D(\lambda, \mu) \leqslant \inf_{x \in \mathrm{D}} f(x) = P^\star \,.$$

Equality, known as strong duality relation requires some additional assumptions.

Lagrangian duality

Strong duality holds under

- convexity of the problem
- constraint qualifications

A simple way to have constraint qualification (sufficient but not necessary)

Slater's conditions

There is some strictly feasible point $x \in \mathbb{R}^d$ such that

$$\begin{aligned} h_i(x) &= 0 \quad \text{for all } i = 1, \ldots, p \\ g_j(x) &< 0 \quad \text{for all } j = 1, \ldots, q \end{aligned}$$

Karush-Kuhn-Tucker (KKT) conditions

Assume that (i) $f, g_1, \ldots, g_q$ are **differentiable** and **convex**, (ii) that $h_1, \ldots h_p$ are **affine** functions and that (iii) Slater's condition holds.

Then $x^\star \in \mathbb{R}^d$ is a solution of the primal problem if and only if there is $(\lambda^\star, \mu^\star) \in \mathbb{R}^p \times \mathbb{R}^q_+$ such that

$$\nabla_x \mathcal{L}(x^\star, \lambda^\star, \mu^\star) = \nabla f(x^\star) + \sum_{i=1}^n \lambda_i^\star \nabla h_i(x^\star) + \sum_{j=1}^n \mu_j^\star \nabla g_j(x^\star) = 0\,,$$

with

$$\begin{aligned} h_i(x^\star) &= 0 \quad \text{for any } i = 1, \ldots, p\,, \\ g_j(x^\star) &\leqslant 0 \quad \text{for any } j = 1, \ldots, q\,, \\ \mu_j^\star g_j(x^\star) &= 0 \quad \text{for any } j = 1, \ldots, q\,. \end{aligned}$$

- These are known as the KKT conditions
- The last one is called complementary slackness

Take-home message: Lagrangian duality

If

- primal problem is **convex** and
- constraint functions satisfy the **Slater**'s conditions

then

- **strong duality** holds.

If in addition we have that

- functions $f, g_1, \ldots, g_n$ are **differentiable**

then

- KKT conditions are **necessary and sufficient** for optimality

Back to the linear SVM

In the Hard SVM case, a way of classifying $\mathcal{D}_n$ with maximum margin is to solve the following problem:

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} f(w)\,,$$

under the constraints:

$$g_i(w) \leqslant 0\,,$$

for all $1 \leqslant i \leqslant n$, where

- $f(w) = \|w\|_2^2/2$ is **strongly convex**, since

$$\nabla^2 f(w) = I_d \succ 0$$

- Constraints are $g_i(w, b) \leqslant 0$ with **affine** functions

$$g_i(w, b) = 1 - Y_i(\langle X_i, w\rangle + b)\,.$$

Back to the linear SVM

The KKT conditions allows to obtain the dual formulation of the problem.

Lagragian

- Introduce dual variables $\mu_i \geqslant 0$ for $i = 1, \dots, n$ corresponding to the constraints $g_i(w, b) \leqslant 0$.
- For $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $\mu = (\mu_1, \dots \mu_n) \in \mathbb{R}_+^n$, define the Lagrangian

$$\mathcal{L}(w, b, \mu) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{n} \mu_i \big(1 - y_i(\langle w, x_i \rangle + b)\big) .$$

KKT and linear SVM

$$\mathcal{L}(w, b, \mu) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \mu_i \big(1 - y_i(\langle w, x_i \rangle + b)\big)$$

KKT conditions

Set the gradient to zero

$$\nabla_w L(w, b, \mu) = w - \sum_{i=1}^{n} \mu_i y_i x_i = 0 \quad \text{namely} \quad w = \sum_{i=1}^{n} \mu_i y_i x_i$$

$$\nabla_b L(w, b, \mu) = -\sum_{i=1}^{n} \mu_i y_i = 0 \quad \text{namely} \quad \sum_{i=1}^{n} \mu_i y_i = 0$$

Write the complementary slackness condition: $\forall i = 1, \dots, n$

$$\mu_i \big(1 - y_i(\langle w, x_i \rangle + b)\big) = 0 \quad \text{namely} \quad \mu_i = 0 \ \text{ or } \ y_i(\langle w, x_i \rangle + b) = 1$$

SVM that's the name

At the optimum,

- There are **dual** variables $\mu_i \geqslant 0$ such that the **primal** solution (w, b) satisfies
$$w = \sum_{i=1}^{n} \mu_i y_i x_i$$
- We have that
$$\mu_i \neq 0 \quad \text{iff} \quad y_i(\langle w, x_i \rangle + b) = 1$$

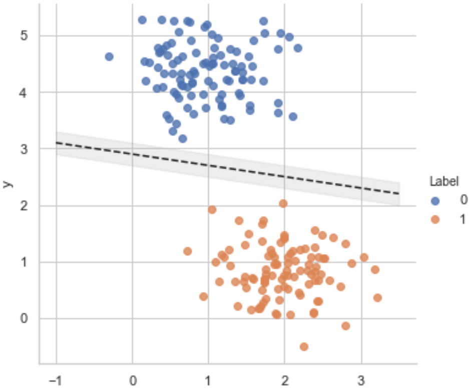
This means that

- w writes as a linear combination of the features vectors x_i that belong to the marginal hyperplanes $\{x \in \mathbb{R}^d : w^T x + b = \pm 1\}$
- These vectors x_i are called support vectors

The support vectors fully define the maximum-margin hyperplane, hence the name **Support Vector Machine**

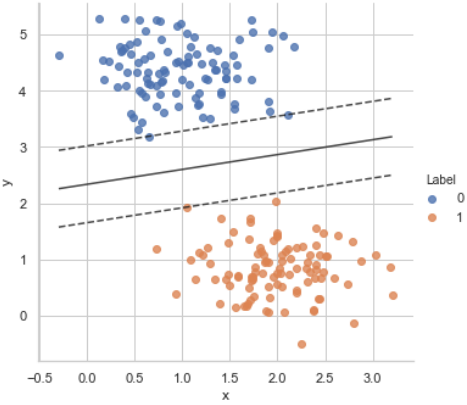
Linearly separable dataset


```
X, y = make_blobs(n_samples = 200, centers = 2, random_state = 0, cluster_std = 0.50)
simulated_data = pd.DataFrame(columns = ["X1","X2","Label"])

simulated_data["x"]     = X[:,0]
simulated_data["y"]     = X[:,1]
simulated_data["Label"] = y

# Use the 'Label' argument to provide a factor variable
sns.set_style("whitegrid")
sns.lmplot(x = "x", y = "y", data = simulated_data, fit_reg = False, hue = 'Label', legend = True)

slope  = 1.1
offset = 0.85
margin = 0.2

xfit = np.linspace(-1, 3.5)
yfit = m * xfit + b

plt.plot(xfit, yfit, '--k')
plt.fill_between(xfit, yfit - d, yfit + d, color = '#AAAAAA', alpha = 0.2)
```



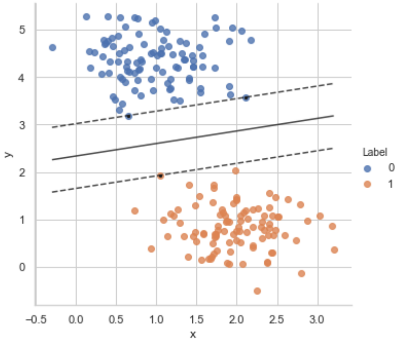
SVM


```
# Classification based on a support vector classifier
model = SVC(kernel='linear', C=10)
model.fit(X, y)
sns.set_style("whitegrid")
sns.lmplot(x = "x", y = "y", data = simulated_data, fit_reg = False, hue = 'Label', legend = True)

xlim  = [np.min(X[:,0]), np.max(X[:,0])]
ylim  = [np.min(X[:,1]), np.max(X[:,1])]
xplot = np.linspace(xlim[0], xlim[1], 30)
yplot = np.linspace(ylim[0], ylim[1], 30)

Yplot, Xplot = np.meshgrid(yplot, xplot)
xy           = np.vstack([Xplot.ravel(), Yplot.ravel()]).T
P            = model.decision_function(xy).reshape(Xplot.shape)

# plot decision boundary and margins
plt.contour(Xplot, Yplot, P, colors = 'k', levels = [-1, 0, 1], alpha = 0.8,
            linestyles = ['--', '-', '--'])
```



Support vectors


```
sns.set_style("whitegrid")
sns.lmplot(x = "x", y = "y", data = simulated_data, fit_reg = False, hue = 'Label', legend = True)
# plot decision boundary and margins
plt.contour(Xplot, Yplot, P, colors = 'k', levels = [-1, 0, 1], alpha = 0.8,
            linestyles = ['--', '-', '--'])
plt.scatter(model.support_vectors_[:, 0], model.support_vectors_[:, 1], s = 5, c = 'k');
```



Dual optimization problem I

Under strong duality, primal and dual problems are strongly related, and one can be used to solve the other.

- Recall that the Lagrangian is

$$\mathcal{L}(w, b, \mu) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{n} \mu_i \big(1 - y_i(\langle w, x_i \rangle + b)\big)$$

- Plug $w = \sum_{i=1}^{n} \mu_i y_i x_i$ in this equation to obtain

$$\mathcal{L}(w, b, \mu) = \frac{1}{2} \Big\| \sum_{i=1}^{n} \mu_i y_i x_i \Big\|_2^2 + \sum_{i=1}^{n} \mu_i - b \sum_{i=1}^{n} \mu_i y_i - \sum_{i,j=1}^{n} \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle \,.$$

Dual optimization problem II

- Recalling that $\sum_{i=1}^{n} \mu_i y_i = 0$ and doing some algebra provides the dual formulation.

Dual formulation

The dual problem amounts to solve:

$$\max_{\mu \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \mu_i - \frac{1}{2} \sum_{i,j=1}^{n} \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle ,$$

under the constraints

$$\mu_i \geqslant 0 \quad \text{and} \quad \sum_{i=1}^{n} \mu_i y_i = 0 \quad \text{for all} \quad i = 1, \ldots, n \, .$$

Comments

- As in the primal formulation, it is again a quadratic programming problem.
- At optimum, we have (using KKT conditions) that the decision function is expressed using the dual variables as

$$x \mapsto sign(w^\top x + b) = sign\left(\sum_{i=1}^{n} \mu_i y_i \langle x, x_i \rangle + b\right)$$

- The intercept b can be expressed for any support vector x_i as

$$b = y_i - \sum_{j=1}^{n} \mu_j y_j \langle x_i, x_j \rangle$$

About the margin

This allows to write the margin as a function of the dual variables

- Multiplying the last equality by $\mu_i y_i$ and summing entails

$$\sum_{i=1}^{n} \mu_i y_i b = \sum_{i=1}^{n} \mu_i y_i^2 - \sum_{i,j=1}^{n} \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle$$

- Namely recalling that at optimum $\sum_{i=1}^{n} \mu_i y_i = 0$ and $w = \sum_{i=1}^{n} \mu_i y_i x_i$ we get

$$0 = \sum_{i=1}^{n} \mu_i = \|w\|_2^2 , \qquad \text{namely}$$

$$\text{margin} = \frac{1}{\|w\|_2^2} = \frac{1}{\sum_{i=1}^{n} \mu_i} = \frac{1}{\|\mu\|_1}$$

Relaxation for the non-separable case

$\rightharpoondown$ Restricting the problem to linearly separable training data sets is a somehow strong assumption.

$\rightharpoondown$ Inequality constraints in the quadratic optimization problem can be relaxed.

Replace the constraints

$$Y_i(\langle w, X_i \rangle + b) \geqslant 1 \quad \text{for all} \quad i = 1, \ldots, n,$$

Relaxation for the non-separable case

$\rightharpoondown$ Restricting the problem to linearly separable training data sets is a somehow strong assumption.

$\rightharpoondown$ Inequality constraints in the quadratic optimization problem can be relaxed.

Replace the constraints

$$Y_i(\langle w, X_i \rangle + b) \geqslant 1 \quad \text{for all} \quad i = 1, \dots, n,$$

that are too strong, by the **relaxed** ones

$$Y_i(\langle w, X_i \rangle + b) \geqslant 1 - s_i \quad \text{for all} \quad i = 1, \dots, n,$$

for slack variables $s_1, \dots, s_n \geqslant 0$

Linear SVM: non-separable case

The original problem

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|w\|_2^2 \ ,$$

under the constraints

$$Y_i(\langle X_i, w \rangle + b) \geqslant 1 \ \text{ for all } \ i = 1, \ldots, n \, .$$

is replaced by the relaxation using slack variables

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, s \in \mathbb{R}^n} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n s_i \ ,$$

under the constraints

$$Y_i(\langle X_i, w \rangle + b) \geqslant 1 - s_i \ \text{ and } \ s_i \geqslant 0 \ \forall \ i = 1, \ldots, n \, .$$

- The slack $s_i \geqslant 0$ measures the the distance by which x_i violates the desired inequality $Y_i(\langle X_i, w\rangle + b) \geqslant 1$
- A vector x_i with $0 < Y_i(\langle X_i, w\rangle + b) < 1$ is correctly classified but is an outlier, since $s_i > 0$
- If we omit outliers, training data is correctly classified by the hyperplane $\{x \in \mathbb{R}^d : \langle x, w\rangle + b = 0\}$ with a margin $1/\|w\|_2^2$
- The margin $1/\|w\|_2^2$ is called a **soft-margin** (in the non-separable case), while it is a **hard-margin** in the separable case

Linear SVM: non-separable case

Relaxed margin problem

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} s_i ,$$

under the constraints

$$Y_i(\langle X_i, w\rangle + b) \geqslant 1 - s_i \quad \text{and} \quad s_i \geqslant 0 \;\forall\; i = 1, \ldots, n .$$

Once again:

- This problem admits a **unique** solution.
- It is a quadratic programming problem.

The constant $C > 0$ is chosen using V-fold cross-valiation.

Linear SVM: non-separable case

Lagrangian

$$\mathcal{L}(w, b, s, \mu, \beta) = \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} s_i + \sum_{i=1}^{n} \mu_i \big(1 - s_i - y_i(\langle w, x_i \rangle + b)\big) - \sum_{i=1}^{n} \beta_i s_i$$

with $\mu_i \geqslant 0$ and $\beta_i \geqslant 0$.

At optimum:

- set the gradients ∇_w, ∇_b and ∇_s to zero ;
- write the complementary conditions.

Linear SVM: non-separable case

$$\nabla_w L(w, b, s, \mu, \beta) = w - \sum_{i=1}^{n} \mu_i y_i x_i = 0 \quad \text{i.e.} \quad w = \sum_{i=1}^{n} \mu_i y_i x_i$$

$$\nabla_b L(w, b, s, \mu, \beta) = -\sum_{i=1}^{n} \mu_i y_i = 0 \quad \text{i.e.} \quad \sum_{i=1}^{n} \mu_i y_i = 0$$

$$\nabla_s L(w, b, s, \mu, \beta) = C - \mu_i - \beta_i = 0 \quad \text{i.e.} \quad \mu_i + \beta_i = C$$

and the complementary condition

$$\mu_i \big(1 - s_i - y_i(\langle w, x_i \rangle + b)\big) = 0 \text{ i.e. } \mu_i = 0 \;\text{ or }\; y_i(\langle w, x_i \rangle + b) = 1 - s_i$$

$$\beta_i s_i = 0 \quad \text{i.e.} \quad \beta_i = 0 \;\text{ or }\; s_i = 0$$

for all $i = 1, \ldots, n$

Linear SVM: non-separable case

- $w = \sum_{i=1}^{n} \mu_i y_i x_i$
- If $\mu_i \neq 0$ we say that x_i is a support vector and in this case $y_i(\langle w, x_i \rangle + b) = 1 - s_i$.
  - If $s_i = 0$ then x_i belongs to a margin hyperplane.
  - If $s_i \neq 0$ then x_i is an outlier and $\beta_i = 0$ and then $\mu_i = C$.

Support vectors either belong to a marginal hyperplane, or are outliers with $\mu_i = C$

To the dual problem...

- Plugging $w = \sum_{i=1}^{n} \mu_i y_i x_i$ in $L(w, b, s, \mu, \beta)$ leads to the same formula as before

$$\sum_{i=1}^{n} \mu_i - \frac{1}{2} \sum_{i,j=1}^{n} \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle$$

To the dual problem...

- Plugging $w = \sum_{i=1}^n \mu_i y_i x_i$ in $L(w, b, s, \mu, \beta)$ leads to the same formula as before

$$\sum_{i=1}^n \mu_i - \frac{1}{2} \sum_{i,j=1}^n \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle$$

- with the constraints

$$\mu_i \geqslant 0, \quad \beta_i \geqslant 0, \quad \sum_{i=1}^n \mu_i y_i = 0, \quad \mu_i + \beta_i = C$$

  that can be rewritten for as

$$0 \leqslant \mu_i \leqslant C, \quad \sum_{i=1}^n \mu_i y_i = 0$$

  for all $i = 1, \ldots, n$

Dual problem

Dual problem

$$\max_{\mu \in \mathbb{R}^n} \quad \sum_{i=1}^n \mu_i - \frac{1}{2} \sum_{i,j=1}^n \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leqslant \mu_i \leqslant C \quad \text{and} \quad \sum_{i=1}^n \mu_i y_i = 0 \quad \text{for all} \quad i = 1, \ldots, n$$

- This is the same problem as before, but with the extra constraint

$$\mu_i \leqslant C$$

- It is again a convex quadratic program

Linear SVM: non-separable case

As in the linearly separable case, the label prediction is expressed using the dual variables.

Labels given by

$$x \mapsto sign(w^T x + b) = sign\left(\sum_{i=1}^{n} \mu_i y_i \langle x, x_i \rangle + b\right)$$

The intercept b can be expressed for a support vector x_i such that $0 < \mu_i < C$ as

$$b = y_i - \sum_{j=1}^{n} \mu_j y_j \langle x_i, x_j \rangle$$

An important remark

The dual problem

$$\max_{\mu \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \mu_i - \frac{1}{2} \sum_{i,j=1}^{n} \mu_i \mu_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leqslant \mu_i \leqslant C \ \text{ and } \ \sum_{i=1}^{n} \mu_i y_i = 0 \ \text{ for all } \ i = 1, \ldots, n$$

and the label prediction (using dual variables)

$$x \mapsto sign(w^T x + b) = sign\left(\sum_{i=1}^{n} \mu_i y_i \langle x, x_i \rangle + b \right)$$

depends only on the features x_i via their **inner products** $\langle x_i, x_j \rangle$!

- This will be particularly important later: **kernel methods**

SVM and the hinge loss

Going back to the primal problem

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, s \in \mathbb{R}^n} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} s_i$$

$$\text{subject to} \quad y_i(\langle x_i, w \rangle + b) \geqslant 1 - s_i \;\; \text{and} \;\; s_i \geqslant 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

Going back to the primal problem

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n s_i$$

subject to $y_i(\langle x_i, w\rangle + b) \geqslant 1 - s_i$ and $s_i \geqslant 0$ for all $i = 1, \ldots, n$

We remark that it can be rewritten as follows.

Reformulation of the primal problem

$$\operatorname{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \max\Big(0, 1 - y_i(\langle x_i, w\rangle + b)\Big).$$

SVM and the hinge loss

The hinge loss function

$$\ell(y, y') = \max(0, 1 - yy') = (1 - yy')_+,$$

the problem can be written as

Reformulation of the primal problem

$$\operatorname{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} \ell(y_i, \langle x_i, w \rangle + b).$$

Leads to an alternative understanding of the linear SVM.

SVM and the hinge loss

Recall that the natural loss is the 0/1 one given by

$$\ell_{0/1}(y, z) = \mathbb{1}_{yz \leqslant 0}.$$

Instead of the Linear SVM, it would be nice to consider

$$\operatorname{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} \mathbb{1}_{y_i(\langle x_i, w \rangle + b) \leqslant 0},$$

but impossible numerically (NP-hard)

Hinge loss is a **convex surrogate** for the 0/1 loss

Conclusion

LDA/QDA

- Model: $X|Y \sim \mathcal{N}$

Logistic regression

- Logistic regression has a nice probabilistic interpretation
- Model $\operatorname{logit}(\mathbb{P}(Y = 1|X)$ is linear in X
- Relies on the choice of the logit link function

✗ does not work on separable dataset

SVM

- No model, only aims at separating points

✓ Thought for separable case

✓ But can be relaxed for the non-separable case