Master in Financial Engineering (EPFL)
Financial Econometrics

Part II: Machine learning and Asset Pricing
Lecture 3: Support Vector Machine—An introduction through linear methods for classification

Florian Pelgrin

EDHEC Business School

February - June 2019

1. Introduction

Main objectives

- Summary of the theory and applications of Support Vector Machine
    - Part I: Review fundamental concepts trough the problem of classification methods, especially using linear methods
        - ✓ Linear regression of an indicator matrix;
        - ✓ Linear Discriminant Analysis (LDA) and some extensions
        - ✓ Logistic regression
        - ✓ Separating Hyperplanes (Perceptron and optimal separation)
        - ✓ A first pass on Support Vector Machine (SVM)

- Part II: Time series predictions using SVM
  - ✓ Linear support vector machines with regressions
  - ✓ Non-separating hyperplanes
  - ✓ Support vector machines with kernels
  - ✓ Applications

1.2. A refresher

- Suppose that one has an outcome measurement (output feature, target) and wishes to predict it based on a set of input features (e.g., some explanatory variables)

- The training set of data is $\left\{\left(x_i^\top, y_i\right), i = 1, \cdots, n\right\}$.

- One implements two statistical methods:

  ✓ Linear regression model;

  ✓ Nearest neighbors.

- **Linear regression model**

$$y_i \quad = \quad \beta_0 + \sum_{i=1}^{p} x_{i,k}\beta_k + u_i$$

where p is the number of input features, u denotes the error term, $x_{i,k}$ is the k-th input feature for observation i, β_0 is the intercept (also known as the *bias* in machine learning),and $\beta_1 \cdots,\beta_p$ are the slope parameters.

The fitted value at the i-th input x_i is

$$\widehat{y}_i \equiv \widehat{y}(x_i) = \tilde{x}_i^\top \hat{\tilde{\beta}}$$

where $\tilde{x}$ includes the constant term and $\tilde{\beta}$ the intercept.
At any arbitrary input x_0, the prediction is:

$$\widehat{y}(x_0) = x_0^\top \hat{\tilde{\beta}}$$

Example: Training data on a pair of inputs and a response variable coded as zero (in blue) and 1 (in orange)

- Step 1: Fit the model
- Step 2: Define a classifier using the fit of the linear regression

$$Y^{\star} = \begin{cases} 1 & \text{if } \widehat{Y} > 0.5 \\ 0 & \text{otherwise.} \end{cases}$$

The set $X^{\top}\widehat{\beta} = 0.5$ is a decision boundary.

- Step 3: Check for misclassification.

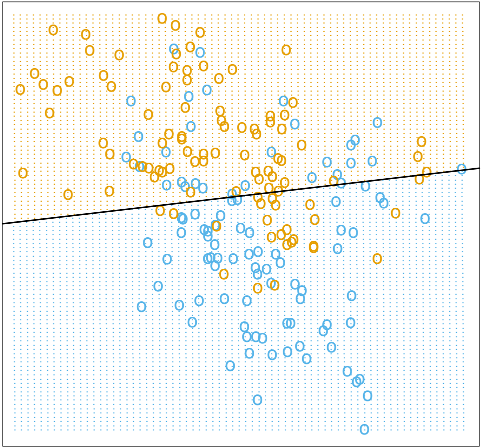

Figure: Linear regression of 0/1 response

Note: Orange (resp., blue) shaded region is the part of the input space classified as "1" (resp., "0"). The line is the decision boundary defined by $x^\top \widehat{\beta} = 0.5$.
Source: The Elements of statistical learning, Hastie et al. (2001).

More generally,

- The training data consists of n pairs $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{a, -a\}$ (say, $a = 1$)

- Define a hyperplane by

$$\left\{x : f(x) = x^\top \beta + \beta_0 = 0\right\}$$

where β is (possibly) a unit vector $\|\beta\| = 1$.

- A classification rule induced by $f(x)$ is

$$G(x) = \texttt{sign}\left[x^\top \beta + \beta_0 + d\right]$$

where d denotes a constant (e.g., d = -0.5 in the example).

- **Nearest neighbors**: Make use of those observations (in the training set) closest in input space x to form the prediction

$$\widehat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i.$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample.

Example: $\widehat{Y}$ is the proportion of orange circles in the neighborhood and it is assigned the value 1 if a majority of neighbors are orange circles.

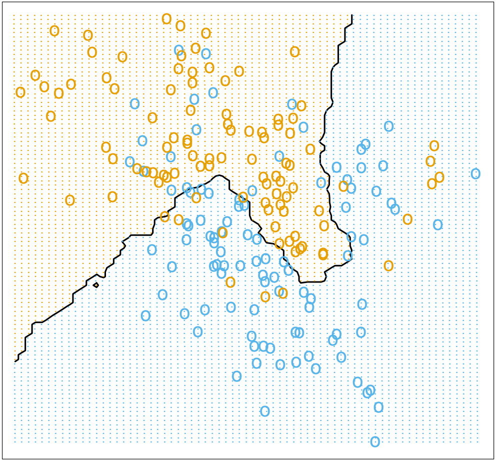

Figure: 15-nearest neighbor averaging

Note: Blue (resp., orange) points, $Y^\star$ correspond to 0 (resp., 1). The class is chosen by majority vote amongst the 15-nearest neighbors.
Source: The Elements of statistical learning, Hastie et al. (2001).

1.3. Framework

- The training data consists of n pairs $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ with $x_i \in \mathbb{R}^p$.
- There are K classes, labeled $1, 2, \cdots, K$
- For each class, inputs are the same and the output is an indicator response variable $\Rightarrow$ there are K indicators Y_k, $k = 1, \cdots, K$, with $Y_k = 1$ if $G = k$ else 0.
- The $n \times K$ *indicator response matrix* Y is $Y = (Y_1, \cdots, Y_K)$.

2. Linear regression for classification

- Using an OLS estimator of the multivariate linear regression model, i.e., fitting a linear regression model to each of the columns of Y (simultaneously), one has

$$\widehat{B} = \left(X^\top X\right)^{-1} X^\top Y$$

where B is a $(p+1) \times K$ coefficient matrix, and X is a $n \times (p+1)$ matrix corresponding to the p inputs and a leading columns of 1's for the intercept.

- Accordingly,

$$\widehat{Y} = X \left(X^\top X\right)^{-1} X^\top Y.$$

Classification rule: Using A new observation with input x

✓ Step 1: Compute the fitted output

$$\widehat{f}(x)^\top = (1, x^\top)\widehat{B}$$

where $\widehat{f} \in \mathbb{R}^K$.

✓ Step 2: Identify the largest component and classify

$$\widehat{G}(x) = \underset{k \in \mathcal{G}}{\texttt{argmax}} \widehat{f}_k(x)$$

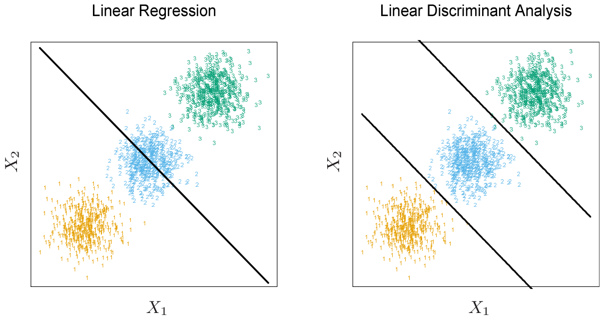

Figure: The masking problem

Source: The Elements of statistical learning, Hastie et al. (2001).

3. Linear and quadratic discriminant analysis

3.1. The decision problem

- The Expected Prediction Error (EPE) is defined by:

$$\texttt{EPE} \quad = \quad \mathbb{E}\left[L(G, \widehat{G}(X))\right]$$

where L denotes the loss function, $\widehat{G}(X)$ the predicted class, and $G = (G_1, \cdots, G_K)$ a discrete set of classes.

- One has

$$\texttt{EPE} \quad = \quad \mathbb{E}_X \sum_{k=1}^{K} L(G_k, \widehat{G}(X))\mathbb{P}(G_k \mid X).$$

and

$$\widehat{G}(X) = \underset{g \in G}{\texttt{argmin}} \sum_{k=1}^{K} L(G_k, \widehat{G}(X))\mathbb{P}(G_k \mid X)$$

- In a case of 0-1 loss function, the minimization problem writes

$$\widehat{G}(X) = \underset{g \in G}{\texttt{argmin}} \left[1 - \mathbb{P}(g \mid X = x)\right]$$

or simply

$$\widehat{G}(X) = G_k \quad \text{if} \quad \mathbb{P}\left(G_k \mid X = x\right) = \max_{g \in G} \mathbb{P}\left(g \mid X = x\right).$$

- Said differently, one classifies to **the most reasonable class** using the conditional distribution $\mathbb{P}(G \mid X)$—a **so-called Bayes classifier**.

3.2. Modeling the conditional distribution

- Using the Bayes theorem:

$$\mathbb{P}\left(G=k \mid X=x\right)=\frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

where

✓ f_k is the class-conditional density of X in class $G=k$;

✓ π_k is the prior probability of class k, with $\sum_{k=1}^{K}\pi_k=1$.

Briefly speaking, having f_k is "almost equivalent" to having the conditional probability (distribution) $\mathbb{P}\left(G=k \mid X=x\right)$

- f_k can be modeled through different class densities:
  - ✓ Gaussian densities : Linear and quadratic discriminant analysis (LDA or QDA);
  - ✓ Flexible mixture of Gaussian densities: nonlinear decision boundaries;
  - ✓ Flexible nonparametric densities: kernel-based approaches;
  - ✓ Naive Bayes models: inputs are conditionally independent in each class, i.e. each of the class densities is the product of marginal densities.

3.3. Linear discriminant analysis

- Suppose that each class density is multivariate Gaussian:

$$f_k(x) = (2\pi)^{-p/2}|\Sigma_k|^{-1/2} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right\}$$

- LDA assumes that classes have a **common covariance matrix** $\Sigma_k = \Sigma$ for all k.

- The decision boundary between two classes k and ℓ can be determined by using the log-ratio:

$$\begin{aligned}\log\left[\frac{\mathbb{P}(G=k \mid X=x)}{\mathbb{P}(G=\ell \mid X=x)}\right] &= \log\left[\frac{f_k(x)\pi_k}{f_\ell(x)\pi_\ell}\right] \\ &= \log\left[\frac{f_k(x)}{f_\ell(x)}\right] + \log\left[\frac{\pi_k}{\pi_\ell}\right]\end{aligned}$$

- Especially,

$$\log\left[\frac{\mathbb{P}(G=k \mid X=x)}{\mathbb{P}(G=\ell \mid X=x)}\right] = \log\left[\frac{\pi_k}{\pi_\ell}\right] - \frac{1}{2}(\mu_k+\mu_\ell)^\top \Sigma^{-1}(\mu_k+\mu_\ell) + x^\top \Sigma^{-1}(\mu_k-\mu_\ell).$$

Remarks:

1. This equation is **linear** in x
2. The assumption "$\Sigma_k = \Sigma$ for all k" greatly simplifies the derivation: neither log of the determinant, nor quadratic term $x^\top \Sigma^{-1} x$ in this expression.

- Consequently, the set defined by

$$\mathbb{P}(G=k \mid X=x) = \mathbb{P}(G=\ell \mid X=x)$$

is linear in x and is an **hyperplane** of dimension p (the number of inputs).

- By dividing $\mathbb{R}^p$ into regions defined by hyperplanes, one obtains a (supervised) classification.

Example:

- Suppose that data are generated by three Gaussian distributions with the same covariance and different means.

- The sample is composed with 30 draws from each Gaussian distribution.

- The linear discriminant functions are defined by:

$$\delta_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log(\pi_k)$$

  and the hyperplane by $\delta_k(x) = \delta_\ell(x)$ for $(k, \ell) = (1, 2), (1, 3)$ and $(2, 3)$.

- The decision rule can also be written $G(x) = \underset{k}{\texttt{argmax}} \delta_k(x)$.

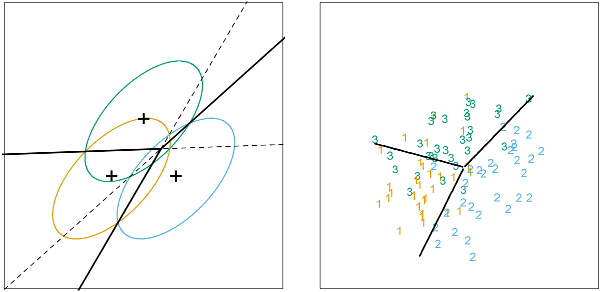

Figure: Linear discriminant analysis

Note: Left panel reports the contours of constant density (covering 95% of the probability). Right panel reports the fitted LDA decision boundaries.
Source: The Elements of statistical learning, Hastie et al. (2001).

Remark: Note that π_k, μ_k and Σ must be estimated...

- An estimate of π_k is:

$$\widehat{\pi}_k = \frac{N_k}{N}$$

  where N_k is the number of class-k observations;

- An estimate of μ_k is:

$$\widehat{\mu}_k = \frac{1}{N_k} \sum_{g_i = k} x_i$$

- An estimate of Σ is:

$$\widehat{\Sigma} = \frac{1}{N - K} \sum_{k=1}^{K} \sum_{g_i = k} (x_i - \widehat{\mu}_k)(x_i - \widehat{\mu}_k)^\top$$

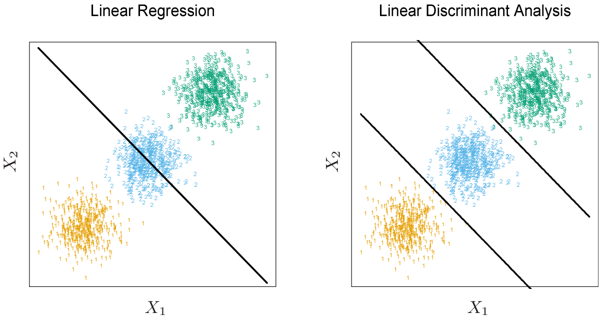

Figure: Back to the masking problem...

Source: The Elements of statistical learning, Hastie et al. (2001).

3.4. Quadratic discriminant analysis

- Suppose now that the Σ_k are not assumed to be equal

- The **quadratic discriminant function** writes

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k) + \log(\pi_k)$$

- The decision boundary between two classes k and ℓ is then given by:

$$\{x : \delta_k(x) = \delta_\ell(x)\}$$

- The estimation proceeds as in the case of LDA, with the exception that separate covariance matrices must be estimated for each class (i.e., curse of dimensionality when p is large!).

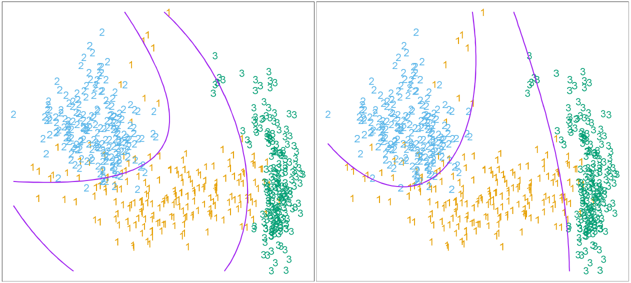

Figure: Quadratic discriminant analysis

Note: Left panel reports quadratic decisions boundaries using LDA with an extension of inputs $(x_1, x_2, x_1^2, x_2^2, x_1 x_2)$. Right panel reports the decision boundaries with QDA. Source: The Elements of statistical learning, Hastie et al. (2001).

3.5. Extensions

Extension 1: Regularized discriminant analysis

- Provide a reasonable solution/compromise regarding the variance-covariance matrice

$$\widehat{\Sigma}_k(\lambda) = \lambda \widehat{\Sigma}_k + (1-\lambda)\widehat{\Sigma}$$

  where $\widehat{\Sigma}$ is the (pooled) covariance matrix with LDA, $\widehat{\Sigma}_k$ is the one with QDA, and λ is a standard regularization parameter with $\lambda \in [0;1]$.

- In general, this leads to decrease the misspecification error (rate) using the training set and the test set.

- **Remark**: A similar treatment can be used for the pooled covariance matrix

$$\widehat{\Sigma}(\alpha) = \alpha \widehat{\Sigma} + (1-\alpha)\sigma^2 \mathrm{I}$$

  where I is the identity matrix and $\sigma^2 \mathrm{I}$ is a spherical covariance matrix.

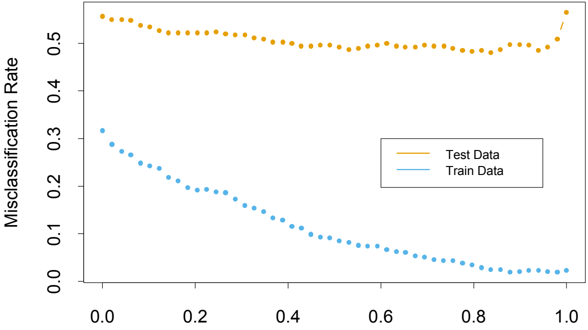

Figure: Regularized discriminant analysis

Extension 2: Reduced-rank or dimension reduction of LDA

- LDA can be viewed as an informative low-dimensional projections of data (like PCA...).

- The LDA problem can be formulated as:

  Can one find a linear combination of the inputs such that the **between-class variance** is maximized relative to the **within-class variance**?

  where the between-class variance is the variance of the class means (resulting from the linear combination), and the within-class is the pooled variance about the means.

- This is a generalized eigenvalue problem

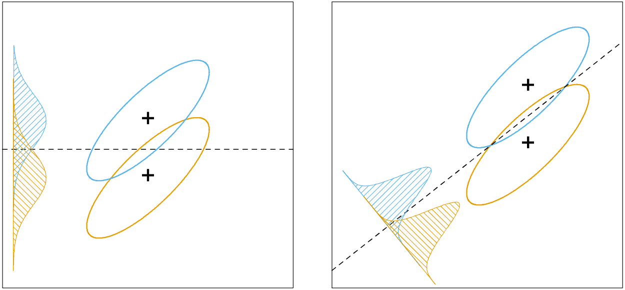

Figure: LDA as a reduced-rank problem

4. Logistic regression

- The posterior (log) odds-ratio is then modeled through linear function

$$
\begin{aligned}
\log\left[\frac{\mathbb{P}(G=1 \mid X=x)}{\mathbb{P}(G=K \mid X=x)}\right] &= \beta_{1,0}+\beta_1^\top x \\
\log\left[\frac{\mathbb{P}(G=2 \mid X=x)}{\mathbb{P}(G=K \mid X=x)}\right] &= \beta_{2,0}+\beta_2^\top x \\
&\vdots \\
\log\left[\frac{\mathbb{P}(G=K-1 \mid X=x)}{\mathbb{P}(G=K \mid X=x)}\right] &= \beta_{K-1,0}+\beta_{K-1}^\top x
\end{aligned}
$$

where the choice of denominator (class K) is arbitrary.

- This is equivalent to

$$
\begin{aligned}
\mathbb{P}(G=k|X=x) &= \frac{\exp\left(\beta_{k,0}+\beta_k^\top x\right)}{1+\sum_{\ell=1}^{K-1}\exp\left(\beta_{\ell,0}+\beta_\ell^\top x\right)} \quad k=1,\cdots,K-1 \\
\mathbb{P}(G=K|X=x) &= \frac{1}{1+\sum_{\ell=1}^{K-1}\exp\left(\beta_{\ell,0}+\beta_\ell^\top x\right)}
\end{aligned}
$$

with $\sum_k \mathbb{P}(G=k|X=x)=1$.

- The hyperplanes are directly defined by the odds-ratio.

- The parameters are fitted by maximizing the conditional likelihood, i.e. the multinomial likelihood with probabilities $\mathbb{P}\left(G = k|X\right)$. Especially, the joint density of $\left(X, G = k\right)$ is given by (in a generic form):

$$\mathbb{P}\left(X, G = k\right) = \mathbb{P}(X)\mathbb{P}(G = k|X)$$

- Notably, the marginal density of X is ignored and can be viewed as being estimated in a nonparametric sense (i.e., the empirical distribution places a mass $1/n$ at each observation.

- In contrast, **LDA** leads to maximize the full log-likelihood, using the joint density

$$\mathbb{P}\left(X, G = k\right) = \phi(X; \mu_k, \Sigma)\pi_k$$

where $\phi(X; \mu_k, \Sigma)$ is the pdf of a multivariate normal distribution with expectation μ_k and covariance matrix Σ, and the marginal density $\mathbb{P}(X)$ is a mixture density

$$\mathbb{P}(X) = \sum_{k=1}^{K} \pi_k \phi(X; \mu_k, \Sigma)$$

that depends on the parameters of interest!

5. Separating hyperplanes

5.1. Example

- Consider 20 points in two classes in $\mathbb{R}^2$.
- Linear regression classifier:
  - ✓ Regress the -1/1 Y response on $X = (X_1, X_2)$ (with an intercept):
    $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + u$$
  - ✓ Classification rule:
    $$G(x) = \texttt{sign}\left[x^\top \beta + \beta_0\right]$$

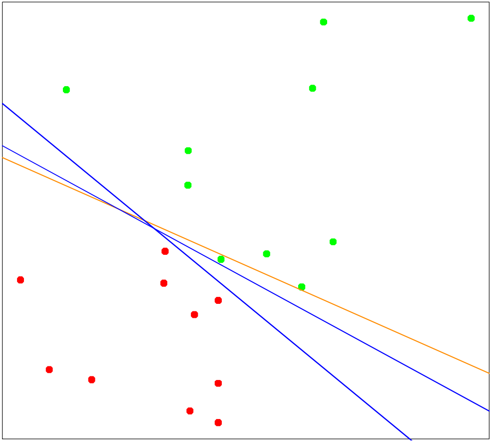

Figure: Separable classes

Note: The orange line provides the least squares solution whereas the blues lines other boundaries.

- Classifiers, which are defined by a linear combination of the features and return the sign, are called perceptrons (Rosenblatt, 1958).

- Such classifiers provide the foundations for the neural network models (see Lecture 5).

5.2. Perceptron learning algorithm

- Objective: Find a separating hyperplane by minimizing the distance of misclassified points to the decision boundary

- Questions:

  Q1. When are points misclassified in our example?
  Answer: A response $y_i = 1$ is misclassified when

  $$x_i^\top \widehat{\beta} + \widehat{\beta}_0 < 0$$

  A response $y_i = -1$ is misclassified when

  $$x_i^\top \widehat{\beta} + \widehat{\beta}_0 > 0$$

  i.e. when the sign is wrongly predicted.

Q2. How to measure the distance to the decision boundary? Back to geometry and algebra...

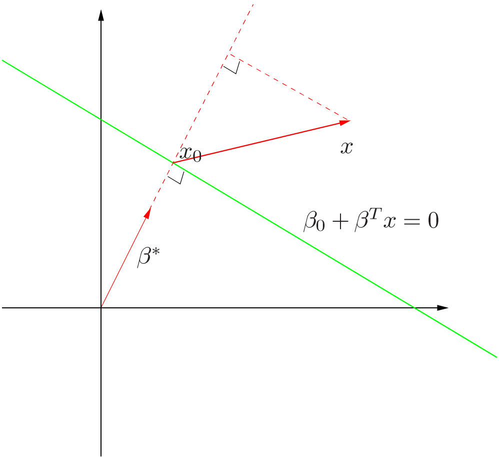

Figure: Distance to a separating hyperplane

- Consider an hyperplane (affine) plan defined by

$$\mathcal{L} = \left\{ f(x) \equiv \beta_0 + \beta^\top x = 0 \right\}$$

This is a line defined by $\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$ with $x = (x_1, x_2)$.

- Then
  - For two points $z_1, z_2 \in \mathcal{L}$, $\beta^\top (z_1 - z_2) = 0$ and the vector normal to L is $\beta^\star = \frac{\beta}{\|\beta\|}$;
  - For any point x_0, $\beta^\top x_0 = -\beta_0$;
  - The **signed distance** (and not the distance!) between $x \in \mathcal{L}^c$ and $x_0 \in \mathcal{L}$ is

$$\underbrace{\beta^{\star\top}(x - x_0)}_{\text{inner product}} = \underbrace{\frac{1}{\|\beta\|}}_{=\frac{1}{\|f'(x)\|}} \underbrace{\left(\beta^\top x + \beta_0\right)}_{=f(x)}.$$

  - The **distance** of interest is then:

$$-y_i \left(x_i^\top \beta + \beta_0 \right)$$

- The objective function (piecewise linear function) to minimize is:

$$D(\beta, \beta_0) = -\sum_{i \in \mathcal{M}} y_i \left(x_i^\top \beta + \beta_0 \right)$$

where $\mathcal{M}$ denotes the set of misclassified points.

- Parameters can be then estimated using a *stochastic gradient descent*:

$$\begin{pmatrix} \beta^{(s)} \\ \beta_0^{(s)} \end{pmatrix} \leftarrow \begin{pmatrix} \beta^{(s-1)} \\ \beta_0^{(s-1)} \end{pmatrix} - \rho \begin{pmatrix} -y_i x_i \\ -y_i \end{pmatrix}$$

where ρ is the learning rate and $\begin{pmatrix} -y_i x_i \\ -y_i \end{pmatrix}$ is the gradient of the objective function for (x_i, y_i).

Key issues:

- Data are not always separable as in Perceptron!
- When data are separable: many solutions might exist and it depends on the starting values.
- The number of iterations to achieve convergence can be quite large!
- In the presence of non separable data, convergence will not occur but this is difficult to assess (occurrence of cycles that are long to detect).

5.3. Optimal separating hyperplanes

- Suppose that there are two classes
- Objective: Separate two classes and maximize the distance to the closest point from either class (Vapnik, 1996)
- How? Using the training sample,
  - Find a maximum **margin** separating two classes;
  - This requires **support points** that lie on the boundary of the margin (no training point being inside the margin);
  - The optimal separating hyperplane bisects the region induced by the maximum margin;

Remark: Note that some points might be inside the "margin" when using the test sample.

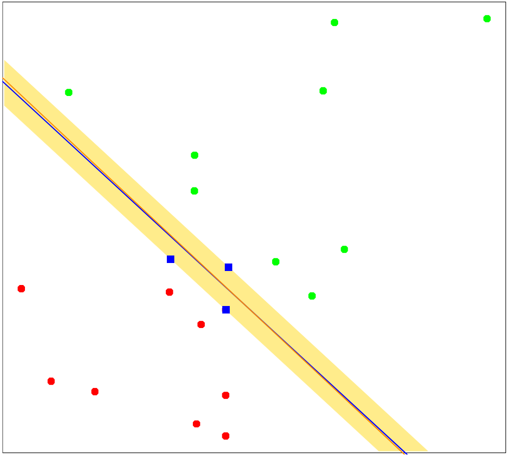

Figure: Optimal separating hyperplanes

- The maximization problem involves
  - A margin, i.e. a signed distance, say M;
  - To impose that some (training) points either lie on the boundary of the margin or do not belong to the margin:

$$y_i \left(x_i^\top \beta + \beta_0 \right) \geq M$$

  for $i = 1, \cdots, n$.

- Consequently, the optimization problem writes

$$\begin{aligned} &\max_{\beta, \beta_0, \|\beta\|=1} \quad M \\ &\text{s.t.} \quad y_i \left(x_i^\top \beta + \beta_0 \right) \geq M \quad \text{for } i = 1, \cdots, n. \end{aligned}$$

- This problem is equivalent to:

$$\min_{\beta,\beta_0} \quad \frac{1}{2}\|\beta\|^2$$
$$\text{s.t.} \quad y_i\left(x_i^\top \beta + \beta\right) \geq 1 \quad \text{for } i = 1, \cdots, n.$$

- This is a convex optimization problem that can be solved with either the primal approach (the Generalized Lagrangian function) or the dual approach (so-called Wolfe dual).

- The corresponding Kuhn-Tucker conditions leads to two cases: (1) x_i is on the boundary of the slab; (2) x_i lies outside the slab.

6. Support vector machine: A first pass

6.1. Overview

- Optimal separating hyperplanes: Classes are linearly separable
- But what happens when classes overlap, i.e. classes are nonseparable?
- A first solution is to determine **nonlinear boundaries** by using a linear boundary on a transformed version of the feature space: the so-called **support vector machine problem**.
- Remark: A second set of solutions is to generalize the linear discriminant analysis: the so-called flexible discriminant analysis.

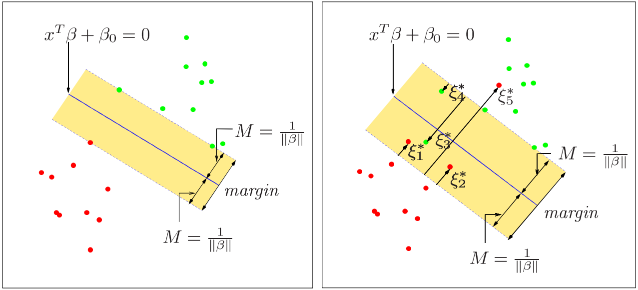

Figure: Separable and nonseparable classes

6.2. SVM classifier

Intuition

- Since the classes overlap in feature space, they cannot be separate!
- But one can still determine a maximum margin and allow for some point to be inside the margin and especially **to be on the wrong side of the margin**...
- The inequality constraints must be redefined to take into account the overlap.
- Especially, one cannot use :

$$y_i \left(x_i^\top \beta + \beta \right) \geq M$$

$\Rightarrow$ Need to modify M...in an additive way or a multiplicative way

- How to measure the overlap?
  - ✓ As the distance from the margin ($i = 1, \cdots, n$)

$$y_i \left(x_i^\top \beta + \beta_0 \right) \geq M - \xi_i$$

    where $\xi = (\xi_1, \cdots, \xi_n)$ is vector of slack variables.
    - ★ This distance is quite natural!
    - ★ But it leads to a nonconvex optimization problem
  - ✓ As a relative distance

$$y_i \left(x_i^\top \beta + \beta_0 \right) \geq M(1 - \xi_i)$$

    - ★ The value ξ_i is the proportional amount by which the prediction $f(x_i) = x_i^\top \beta + \beta_0$ is on the wrong side of its margin;
    - ★ The total proportional amount by which all predictions are on the wrong side of their margin is $\sum_{i=1}^{n} \xi_i$;
    - ★ This total proportional amount can be bounded!
    - ★ Misclassification occurs when $\xi > 1$

- The optimization problem writes:

$$
\begin{aligned}
\min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 \\
\text{s.t.} \quad & y_i\left(x_i^\top \beta + \beta\right) \geq 1 - \xi \quad \text{for } i = 1, \cdots, n \\
& \xi_i \geq 0, \sum_{i=1}^{n} \xi_i \leq \text{constant.}
\end{aligned}
$$

or

$$
\begin{aligned}
\min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 + c\sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad & y_i\left(x_i^\top \beta + \beta\right) \geq 1 - \xi \\
& \xi_i \geq 0 \quad \text{for } i = 1, \cdots, n.
\end{aligned}
$$

where the (implicit) cost parameter c (tuning parameter) replaces the constant term.

Remark: The separable case corresponds to $c = \infty$.

6.3. SVM as a penalization method

- The support vector machine method can be interpreted as a Penalization method
- The corresponding optimization problem writes

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2$$

where $[1 - y_i f(x_i)]_+$ indicates the positive part of $1 - y_i f(x_i)$ and λ is the penalty parameter.

- Remarks:
  1. One can show that $\lambda = 1/C$;
  2. $L(y, f) = [1 - y_i f(x_i)]_+$ is the so-called "hinge loss function"

6.4. SVM for linear regression

- When the response variable is quantitative, the minimization problem can be written:

$$\sum_{i=1}^{n} V\epsilon\left(y_i - f\left(x_i\right)\right) + \frac{\lambda}{2}\|\beta\|^2$$

where

$$f\left(x_i\right) = x_i^\top \beta + \beta_0$$

and $V\epsilon$ is an "ϵ-intensive" error measure

$$V\epsilon(z) = \begin{cases} 0 & \text{if } |z| < \epsilon \\ |z| - \epsilon & \text{otherwise} \end{cases}$$

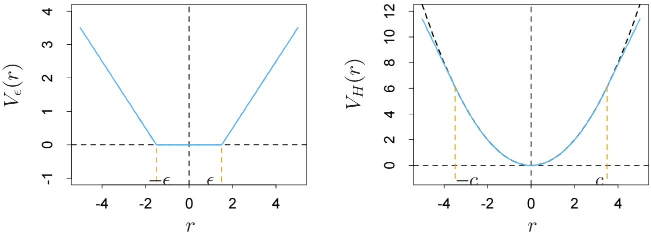

Figure: SVM for regression