08 SVM

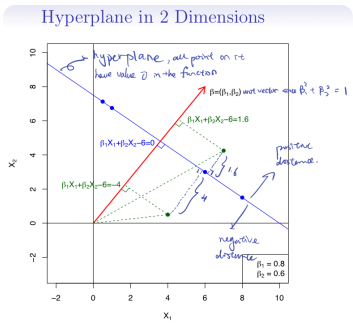
(1) Seperating Hyperplane.

A hyperplane: $\beta_0 + x^T\beta = 0$

⇒ points on one side will be negative; points on the other side will be positive.

⇒ A classification rule:

$$G(x) = \text{sign}[\beta_0 + x^T\beta]$$



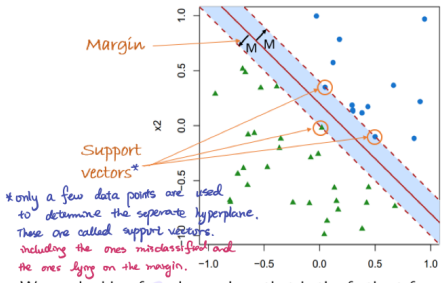
⇒ The best hyperplane is decided by Maximal Margin Classifier (MMC).

$$\begin{aligned}&\underset{\beta_0,\beta_1,\dots,\beta_p}{\text{maximize}}\ M\\ &\text{subject to } \textstyle\sum_{j=1}^p \beta_j^2 = 1,\\ &y_i(\beta_0 + x_i^T\beta) \geq M \text{ for all } i=1,\dots,n\end{aligned}$$

(i.e. all the points are at M distance from the hyperplane)

can be rewritten as:

$$\begin{aligned}&\text{maximise } C\\ &\text{subject to}\\ &\frac{y_i}{\|\beta\|}(\beta_0 + x_i^T\beta) \geq C,\ i=1,\dots,n\end{aligned}$$



* only a few data points are used to determine the seperate hyperplane. These are called support vectors. including the ones misclassified and the ones lying on the margin.

We are looking for a hyperplane that is the farthest from the training observations, i.e. has the largest **margin**.

⇔ maximise $\frac{1}{\|\beta\|}$ subject to $y_i(\beta_0 + x_i^T\beta) \geq 1,\ i=1,\dots,n$

⇐⇒ minimise $\|\beta\|^2$ subject to $y_i(\beta_0 + x_i^T\beta) \geq 1,\ i=1,\dots,n$

Using Lagrangian equations: $\beta_j = \sum_{i=1}^n \alpha_i y_i x_{ij}$ (4) for $j=1,\dots,p$ and $\sum_{i=1}^n \alpha_i y_i = 0$ (5)

⇒ maximise $L = \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_{i'} \alpha_i\alpha_{i'} y_i y_{i'} x_i^T x_{i'}$ (6) subject to $\alpha_i \geq 0$; $\alpha_i[y_i(x_i^T\beta + \beta_0) - 1] = 0$ for $i=1,\dots,n$

note: if $\alpha_i > 0$, then $y_i(x_i^T\beta + \beta_0) = 1$ ⇒ observation i is active.
if $\alpha_i = 0$, then $y_i(x_i^T\beta + \beta_0) > 1$ ⇒ observation i is not active.

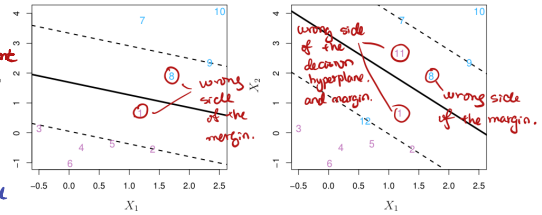
⇒ β_j only depends on active observations, i.e. $\beta_j = \sum_{\text{active } i} \alpha_i x_{ij} y_i$

(2) Support Vector Classifier (Soften seperating hyperplane)

Classification rule: $G(x) = \text{sign}(x^T\hat\beta + \hat\beta_0)$.

Soften MMC: (essentially a regularization method, so its important to standardise all points first).

$$\begin{aligned}&\underset{\beta_0,\beta_1,\dots,\beta_p,\varepsilon_1,\dots,\varepsilon_n}{\text{maximise}}\ M \text{ subject to}\\ &\textstyle\sum_{j=1}^p \beta_j^2 = 1 \text{ (except for } \beta_0),\\ &y_i(\beta_0 + x_i^T\beta) \geq M(1-\varepsilon_i) \text{ → allow some points to be misclassified}\\ &\varepsilon_i \geq 0;\ \textstyle\sum_{i=1}^n \varepsilon_i \leq t \text{ (budget tuning parameter)} \text{ → the more points included, the more stable the model is.}\end{aligned}$$

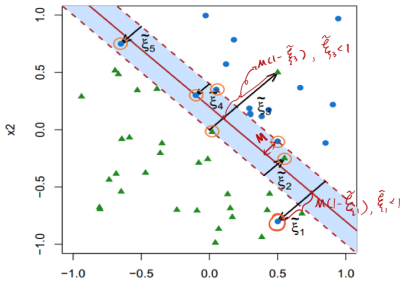


⇓

$$\begin{aligned}&\text{minimise } \|\beta\|^2 \text{ subject to}\\ &y_i(x_i^T\beta + \beta_0) \geq 1 - \varepsilon_i,\\ &\varepsilon_i \geq 0,\ \textstyle\sum_{i=1}^n \varepsilon_i \leq t.\end{aligned}$$

⇓

$$\begin{aligned}&\text{minimise } \tfrac{1}{2}\|\beta\|^2 + \gamma\textstyle\sum_{i=1}^n \varepsilon_i \text{ subject to} \rightarrow \text{can use CV to choose, } \begin{cases}\gamma\uparrow \text{ margin}\downarrow\\ \gamma\downarrow \text{ margin}\uparrow\end{cases}\\ &y_i(x_i^T\beta + \beta_0) \geq 1 - \varepsilon_i,\\ &\varepsilon_i \geq 0\end{aligned}$$



Note that in the above diagram, the margin is $C = 1/\|\beta\|$, so the distances ξ_i marked on the diagram equal $\xi_i/\|\beta\|$.

(3) Support Vector Machine. (enrich the selection space other than linear boundaries).

Classification rule: $G(x) = \text{sign}\left(\beta_0 + \sum_{i\in S} \alpha_i k(x, x_i)\right)$ → set of support vector.

where $k(x, x')$ is known as SVM kernels, and it can be

$k(x, x') = (C + \langle x, x'\rangle)^d$ (Degree d polynomials).

$k(x, x') = \exp(-\gamma\|x - x'\|^2/C)$ (Radial basis function)

$k(x, x') = \sqrt{\{1 + \exp(-C_1\langle x, x'\rangle + C_2)\}}$ (Sigmoid function)

and $\alpha_i = \alpha_i y_i$ (recall that $\beta_j = \sum_{i=1}^n \alpha_i x_{ij} y_i$ ⇒ $\sum_{j=1}^p \beta_j x_j = \sum_{j=1}^p \sum_{i=1}^n \alpha_i y_i x_j = \sum_{i=1}^n \alpha_i \sum_{j=1}^p x_j x_{ij} = \sum_{i=1}^n \alpha_i \langle x_i, x\rangle$ where $\alpha_i = \alpha_i y_i$)

↳ can be $k(x_i, x)$.

So $\alpha_i = 0 \iff \alpha_i = 0 \iff$ observation i is non-active and vice versa.

the optimisation expression is:

minimise $\|\beta\|^2 + \gamma \sum_{i=1}^n \varepsilon_i$ subject to → penalisation on complex model. { $\gamma\uparrow$ wiggly model; $\gamma\downarrow$ simple model.

$Y_i\{h(x_i)^T\beta + \beta_0\} \geq 1 - \varepsilon_i$, $\varepsilon_i \geq 0$

where $h(x_i)$ is the basic function.

* need to do CV on γ as well as parameters in the kernel, e.g. C in radial function.

Again we can use Lagrangian equation, the 2 most important expressions are:

(6) → $L = \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_{i'} \alpha_i\alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'})\rangle$

(4) becomes $\beta_j = \sum_{i=1}^n \alpha_i y_i h(x_i)$ ⇒ $h(x)^T\beta + \beta_0 = \sum_j h(x_j)\beta_j + \beta_0 = \sum_j h(x_j)\sum_i \alpha_i y_i h(x_i) + \beta_0 = \sum_i \alpha_i y_i \langle h(x), h(x_i)\rangle + \beta_0$

where $\langle h(x), h(x')\rangle = K(x, x')$ ⇒ so we can calculate $\langle x, x'\rangle$ first and put it into $K(x, x')$ without having to calculate $\langle h(x), h(x')\rangle$. $K(x, x')$ is a symmetric and positive function

e.g. $K(x, x') = (C + \langle x, x'\rangle)^2$

(4) SVM and penalised loss function.

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

$[1 - Y_i\{\cdots\}]_+$: hinge loss; $\lambda\|\beta\|^2$: ridge penalty. (shrinking towards 0).

is equivalent to SVM

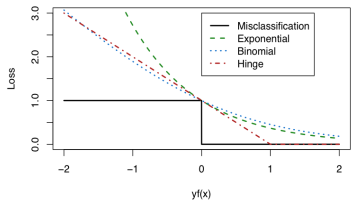
⇒ SVM can be thought of as fitting a basic expansion model $f(x_i) = h(x_i)^T\beta + \beta_0$ to predict Y_i, using hinge loss $(1 - Y_i f(x_i))_+$ and a ridge penalty.

We may add the following to the family of loss functions for binary classification. Here $Y_i \in \{-1, 1\}$ and $f(X_i)$ are predictions with $\text{sign}\{f(X_i)\}$ being the classifier.

- Misclassification loss: $\mathbb{I}[Y_i \neq \text{sign}\{f(X_i)\}]$.
- Binomial loss: $-\log\left\{\frac{1}{1+e^{-2Y_i f(X_i)}}\right\}$.
- Exponential loss: $\exp\{-Y_i f(X_i)\}$.
- SVM loss: $[1 - Y_i f(X_i)]_+$.
- Squared loss: $\left\{Y_i - \frac{e^{2f(X_i)}}{1+e^{2f(X_i)}}\right\}^2$.

For a comprehensive discussion of the different loss functions see pp.426-428 in the book "Elements of statistical learning".

We can view the various loss functions against $Y_i f(X_i)$:



We give results for a variety of binary classification, with performance. Most models have standard error of around 0.5% on the misclassification rate

Model	Misclass. Rate
Single tree	10.4%
Bagged trees	8.4%
Logistic regression*	8.3%
Random forests	5.6%
Adaboost	5.5%
GBM	5.9%
Neural Net	6%
SVM	7.8%

(5) SVM in regression.

$$\sum_{i=1}^n L\{Y_i, h(x_i)^T\beta + \beta_0\} + \frac{\lambda}{2}\|\beta\|^2.$$

In principle any loss function $L(\cdot)$ could be used.

e.g. $L_\varepsilon\{y, f(x)\} = \begin{cases} 0 & \text{if } |y - f(x)| < \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases}$ ⇒ there is a margin ε that as long as the observed values lie within, there is no loss. Outside this the loss grows linearly.