SVM, Logistic Regression a practical review

Claire He, 15/11, STATGU4253/STATGR5253

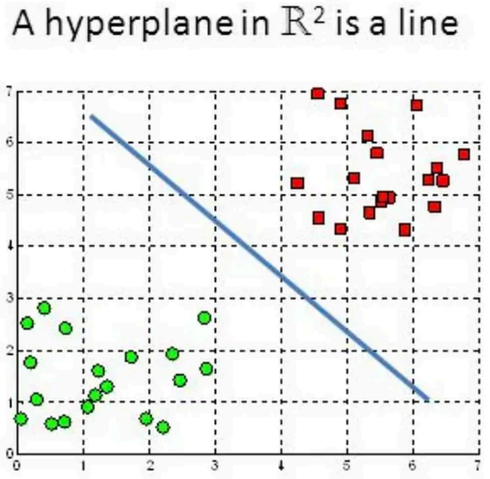
SVM: support vector machine

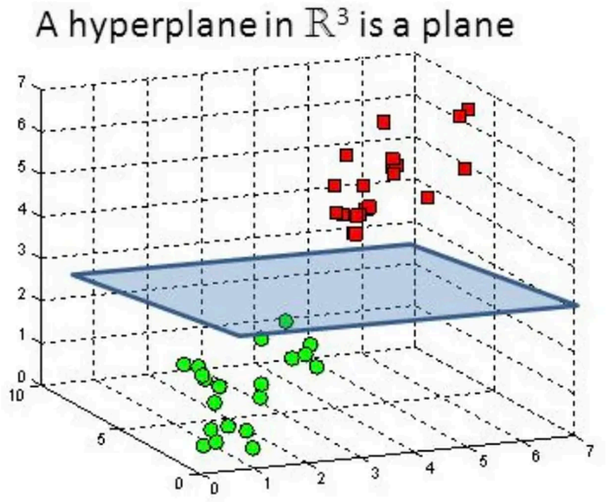
Main ideas

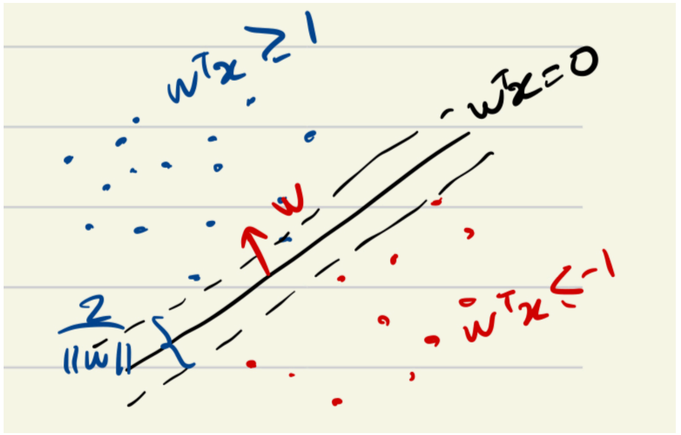
- Find a separating hyperplane (if data is linearly separable). $Y \in \{-1,1\}$, $X \in \mathbb{R}^d$ covariates.
- Maximise the **margin** between the two classes
- Optimization centric problem:

$$\max_{w} \underbrace{\frac{2}{||w||}}_{\text{margin}} \quad \text{s.t.} \quad w^T x_i y_i \geq 1, \ \forall i = 1,\ldots,n$$

—> convex program (quadratic) can be solved! (See additional notes for worked mathematical details)







SVM: support vector machine

Main ideas

- « Support vectors » saturate the constraints —> determine the hyperplane
- When not linearly separable, introduce « slack » (soft constraint) $(\xi_i)_{i=1}^n$ where we allow points on the wrong side within a distance $y_i x_i^T w \geq 1 - \xi_i$
- The value ξ_i is the proportional amount by which the prediction $\hat{y}_i$ is on the wrong side of its margin
- Points well within their boundaries do not matter: points on the margin and within the margin impact the optimization

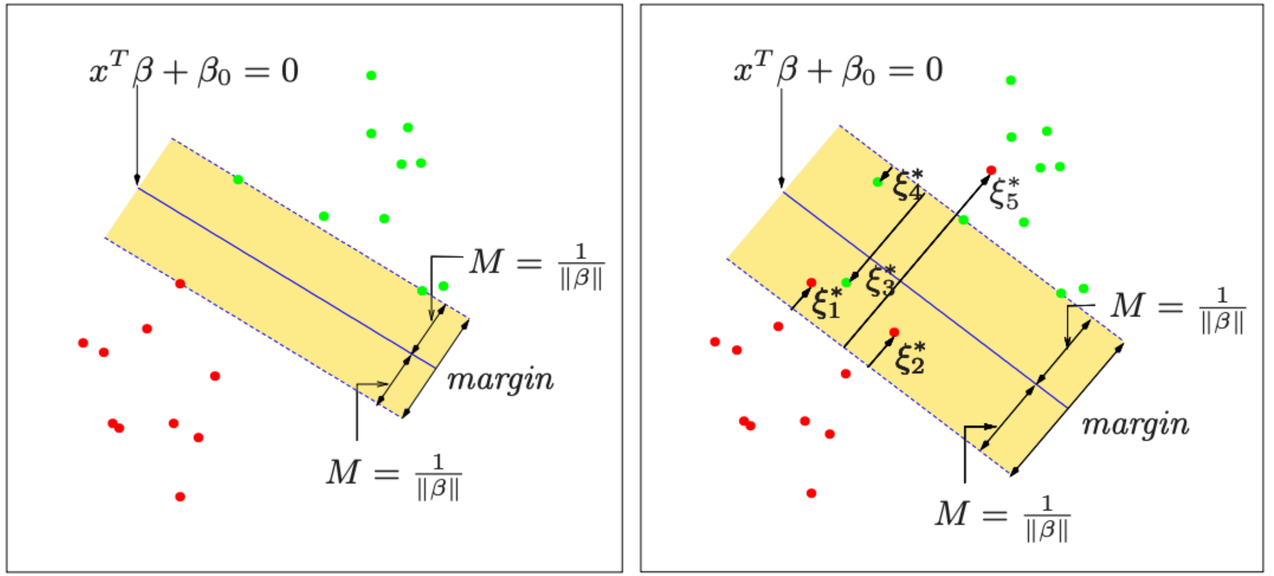
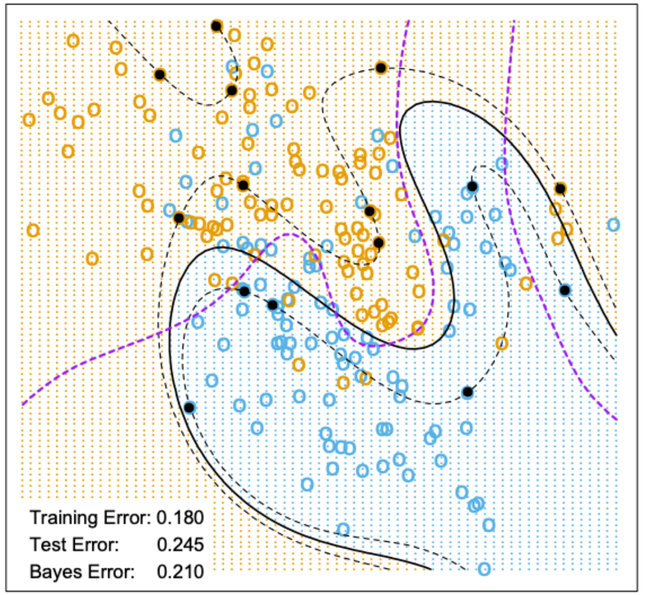


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

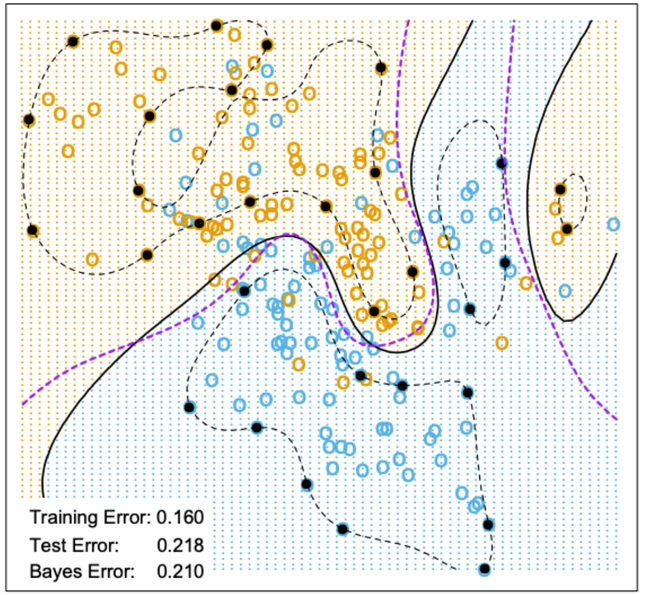
SVM: support vector machines

Kernelisation

- What if there is non-linearity? In the optimization problem, the constraint uses $x_i^T x_j$
- Can kernelise the problem: find a map $\phi(x)$ such that $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ —> « linearisation » in the feature space
- See tutorial in practice here: tutorial SVM
- In practice: can use *sklearn* package



SVM - Radial Kernel in Feature Space

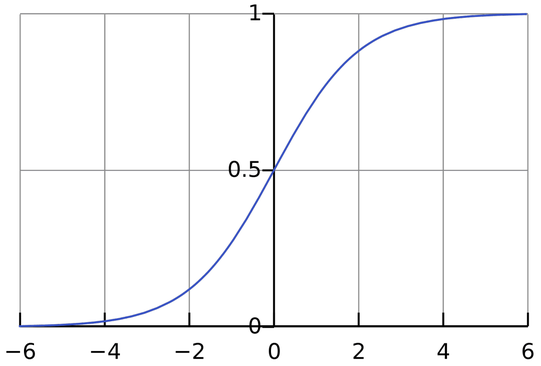


Logistic Regression

Main ideas

Want to model $\log \frac{p}{1-p} = \beta_0 + \beta^T x$ where

$p = P(Y = 1 \mid X = x)$ —> naturally induces a sigmoid transformation



$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{N} \left\{ y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta)) \right\} \\ &= \sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1+e^{\beta^T x_i}) \right\}.\end{aligned}$$

Can be fit using MLE and Newton Raphson algorithm to solve the first order and second order constraints

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i (y_i - p(x_i;\beta)) = 0,$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i {x_i}^T p(x_i;\beta)(1-p(x_i;\beta)).$$

Update til convergence:

$$\beta^{\text{new}} = \beta^{\text{old}} - \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

Logistic Regression

Main ideas

- In practice: can be optimized via gradient descent
  - LR using gradient descent
  - Another LR using gradient descent
  - LR using IRLS (equivalent NR)
- Implementation can be found in *sklearn* packages.