P9120 - Statistical Learning and Data Mining

Lecture 4 - Linear Classification

Min Qian

Department of Biostatistics, Columbia University

September 26, 2024

Outline

The Classification Problem

- Input X, output $Y \in \{1, 2, \ldots, K\}$ (the class label).
- Goal: construct a model $G(X)$ that predicts the class variable using the input variables. The model is called "classifier".
- Loss: $L(Y, G(X)) = 1_{Y \neq G(X)}$.
- Risk: $R(G) = E[L(Y, G(X))] = P(Y \neq G(X))$.
- Optimal classifier (Bayes classifier):

$$G^*(\mathbf{x}) = \arg\min_G R(G) = \arg\max_k (f_k^*(\mathbf{x})),$$

where $f_k^*(\mathbf{x}) = P(Y = k | X = \mathbf{x})$, $k = 1, \ldots, K$.

(- i.e., the class with largest posterior probability.)

Binary Classification using Linear Regression?

$$G^*(\mathbf{x}) = \arg\max_{k \in \{0,1\}} (P(Y = k|X = \mathbf{x})) = 1_{P(Y=1|X=\mathbf{x})>0.5}$$

- Binary outcome: $Y \in \{0, 1\}$.

Multi-class Classification using Linear Regression?

$$G^*(\mathbf{x}) = \arg \max_{Y \in \{1,2,\ldots,K\}} (P(Y = k | X = \mathbf{x}))$$

The Set-up of LDA (Linear Discriminant Analysis)

$$G^*(\mathbf{x}) = \arg \max_{Y \in \{1,2,\ldots,K\}} (P(Y = k | X = \mathbf{x}))$$

- $f_k(\mathbf{x})$: the class specific density of X in class k.
- $\pi_k = P(Y = k)$, the prior probability of class k with $\sum_k \pi_k = 1$.
- By Bayes Theorem,

$$P(Y = k | X = \mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_{l=1}^{K} f_l(\mathbf{x})\pi_l}.$$

- Estimate f_k's and π_k's and choose the class that maximizes $f_k(\mathbf{x})\pi_k$.

Model for the Class Densities $f_k(\mathbf{x})$'s in LDA

- Assume $X|Y = k \sim N_p(\boldsymbol{\mu}_k, \Sigma_k)$, i.e.

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}.$$

  with $\Sigma_k = \Sigma$ for $k = 1, \ldots, p$.

- Choose the class k that maximizes $\log[f_k(\mathbf{x})\pi_k]$,

$$\log[f_k(\mathbf{x})\pi_k] = \delta_k(\mathbf{x}) + \text{ class independent quantity},$$

  where $\delta_k(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$.

Linear Discriminant Functions

- The decision boundary for two classes k and l is $\delta_k(\mathbf{x}) - \delta_l(\mathbf{x}) = 0$. Equivalently,

$$\mathbf{x}^T\Sigma^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l) - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_l)^T\Sigma^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l) + \log\frac{\pi_k}{\pi_l} = 0$$

  which is linear in $\mathbf{x}$.

- The linear discriminant functions

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log\pi_k, k = 1, \ldots, K$$

  give the classification rule

$$G(\mathbf{x}) = \arg\max_k \delta_k(\mathbf{x}).$$

LDA in Practice

In practice parameters of the Gaussian distributions are estimated by

- $\hat{\pi}_k =$
- $\hat{\boldsymbol{\mu}}_k =$
- $\hat{\Sigma} =$

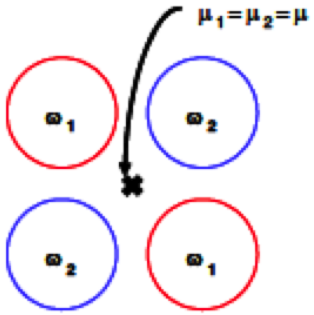
For a new observation with input $\mathbf{x}$, compute

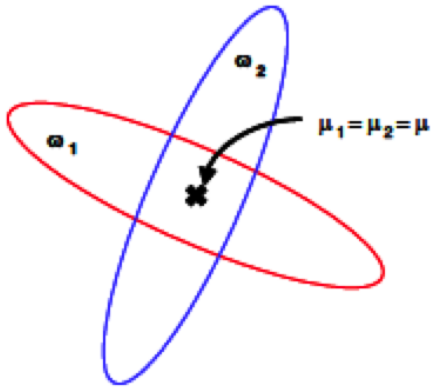
$$\hat{\delta}_k(\mathbf{x}) = \mathbf{x}^T \hat{\Sigma}^{-1} \hat{\boldsymbol{\mu}}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \hat{\Sigma}^{-1} \hat{\boldsymbol{\mu}}_k + \log \hat{\pi}_k, k = 1, \ldots, K.$$

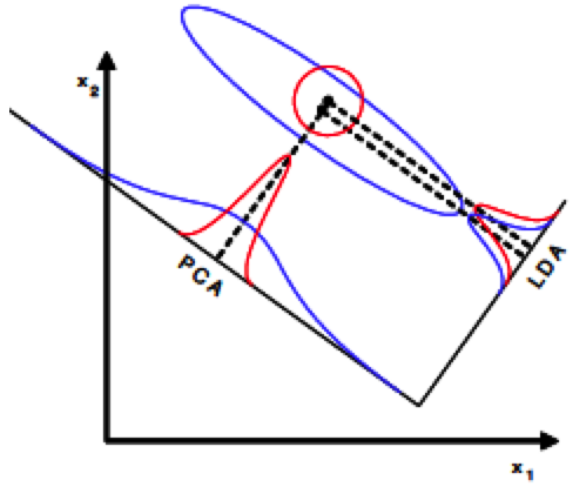
The LDA rule

$$\hat{G}(\mathbf{x}) = \arg\max_k \hat{\delta}_k(\mathbf{x}).$$

Limitation of LDA







Quadratic Discriminant Analysis (QDA)

- Do not assume Σ_k's to be equal across classes.

- Quadratic terms involving $\mathbf{x}$ remain in the discriminant function (called quadratic discriminant function):

$$\delta_k^Q(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) - \frac{1}{2}\log|\Sigma_k| + \log \pi_k.$$

  The classification rule

$$G(\mathbf{x}) = \arg\max_k \delta_k^Q(\mathbf{x}).$$

- QDA is similar to LDA except that Σ_k must be estimated separately for each class:

$$\hat{\Sigma}_k = \sum_{i:y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)/(n_k - 1).$$

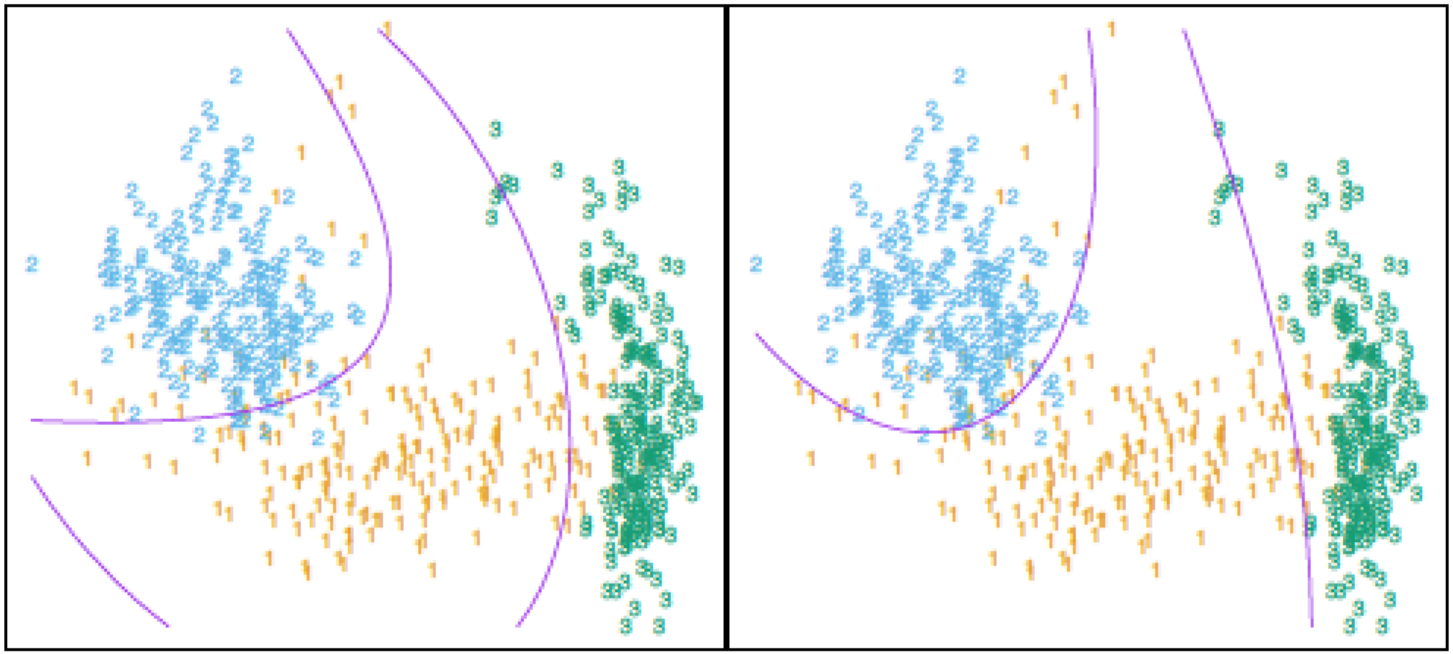


FIGURE 4.6. *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $X_1, X_2, X_1X_2, X_1^2, X_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

Binary Logistic Regression ($Y \in \{0, 1\}$)

- Denote $P(Y = 1|X = \mathbf{x}) \triangleq p(\mathbf{x})$. Model

$$\log \frac{p(\mathbf{x}; \boldsymbol{\beta})}{1 - p(\mathbf{x}; \boldsymbol{\beta})} = \mathbf{x}^T \boldsymbol{\beta} \quad \Rightarrow \quad p(\mathbf{x}; \boldsymbol{\beta}) = \frac{\exp(\mathbf{x}^T \boldsymbol{\beta})}{1 + \exp(\mathbf{x}^T \boldsymbol{\beta})}.$$

- Log-likelihood

$$\begin{aligned} l(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \left\{ y_i \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta})) \right\} \\ &= \sum_{i=1}^{n} \left\{ y_i (\mathbf{x}_i^T \boldsymbol{\beta}) - \log \left(1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta})\right) \right\} \end{aligned}$$

- Parameters are estimated using MLE, denoted as $\hat{\boldsymbol{\beta}}$.

- The classification rule $G(\mathbf{x}) = 1_{\mathbf{x}^T \hat{\boldsymbol{\beta}} > 0}$.

Newton-Raphson Algorithm

Goal: Find solution to

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \left\{ \mathbf{x}_i (y_i - p(\mathbf{x}_i; \boldsymbol{\beta})) \right\} = \mathbf{0}.$$

- Second-order derivative or Hessian matrix

$$\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\sum_{i=1}^{n} \left\{ \mathbf{x}_i \mathbf{x}_i^T p(\mathbf{x}_i; \boldsymbol{\beta})(1 - p(\mathbf{x}_i; \boldsymbol{\beta})) \right\} \text{ is negative semidefinite.}$$

A single Newton update is

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \left(\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \right)^{-1} \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \Bigg|_{\boldsymbol{\beta} = \boldsymbol{\beta}^t}.$$

Gradient Descent

Goal: Find $\boldsymbol{\beta}$ to minimize

$$J(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \Big\{ - y_i(\mathbf{x}_i^T \boldsymbol{\beta}) + \log\left(1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta})\right) \Big\}$$

Gradient Descent:

$$\boldsymbol{\beta}^{t+1} \leftarrow \boldsymbol{\beta}^t - \gamma \frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^t},$$

where γ is the learning rate, and

$$\frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \Big\{ \mathbf{x}_i (p(\mathbf{x}_i; \boldsymbol{\beta}) - y_i) \Big\}.$$

- Gradient descent only uses the first derivative.
- Newton-Raphson uses the second derivative which lead generally faster to a solution if the second derivative is easy to compute.

Multi-Class Logistic Regression

- For $k = 1, \ldots, K-1$, model

$$\log \frac{P(Y = k|X = \mathbf{x})}{P(Y = K|X = \mathbf{x})} = \mathbf{x}^T \boldsymbol{\beta}_k.$$

- A simple calculation shows that

$$P(Y = k|X = \mathbf{x}) = \frac{\exp(\mathbf{x}^T \boldsymbol{\beta}_k)}{1 + \sum_{l=1}^{K-1} \exp(\mathbf{x}^T \boldsymbol{\beta}_l)} = p_k(\mathbf{x}, \boldsymbol{\beta}), \quad k \leq K-1,$$

$$P(Y = K|X = \mathbf{x}) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\mathbf{x}^T \boldsymbol{\beta}_l)} = p_K(\mathbf{x}, \boldsymbol{\beta}).$$

- Recode Y by vector $\mathbf{Y} = (Y_1, \ldots, Y_K)$ with $Y_k = 1_{Y=k}$.
  $\mathbf{Y}|X = \mathbf{x} \sim Multinomial(1, p_1(\mathbf{x}, \boldsymbol{\beta}), \ldots, p_K(\mathbf{x}, \boldsymbol{\beta}))$.

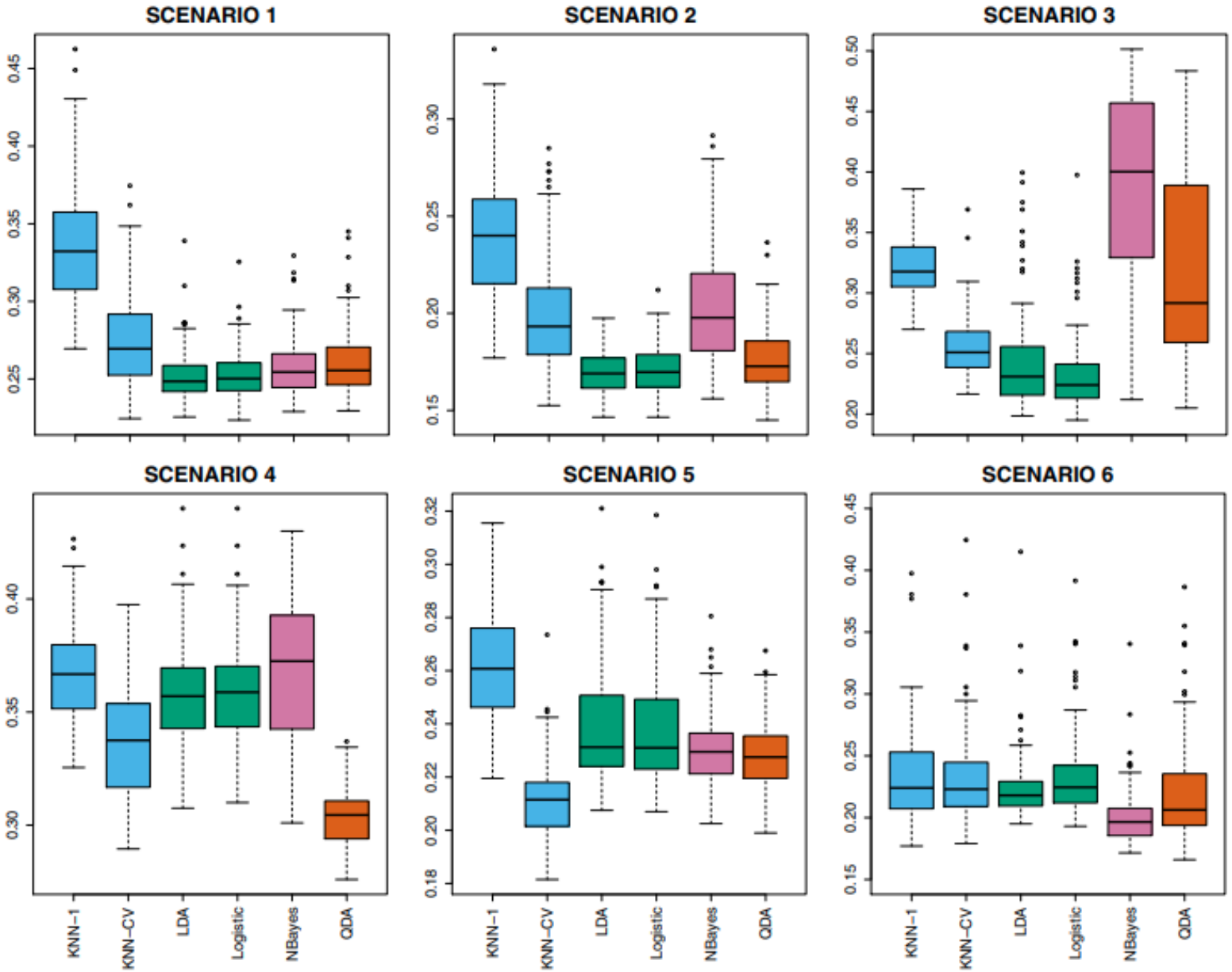
- Log-likelihood:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log p_k(\mathbf{x}_i, \boldsymbol{\beta}).$$

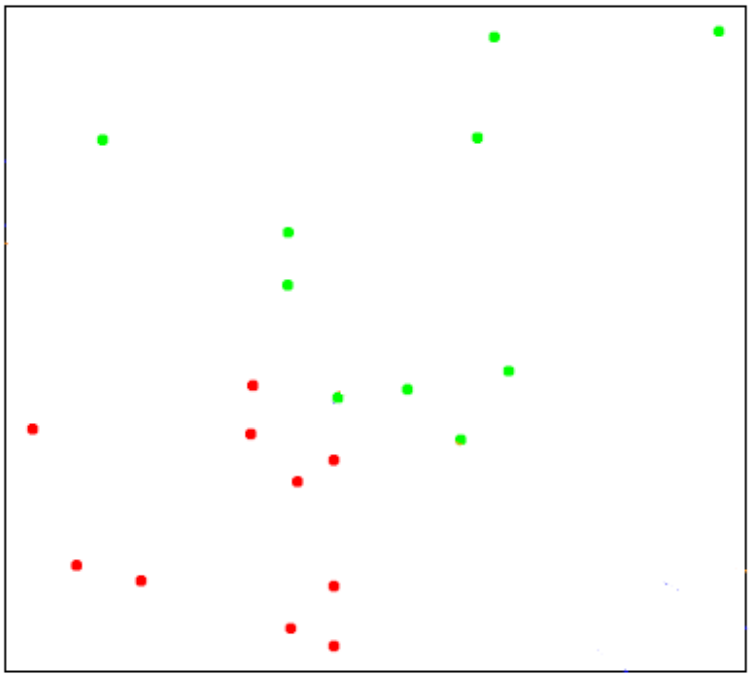
LDA vs. Logistic Regression

- LDA and logistic regression use the same model, but the parameters are estimated differently.
- Logistic regression maximizes conditional likelihood. LDA maximizes the full log-likelihood.
- It is generally felt that logistic regression is a safer, more robust bet than the LDA model, relying on fewer assumptions (bias-variance tradeoff).
- "It is our experience that the models give very similar results, even when LDA is used inappropriately, such as with qualitative predictors." (from [ESL])

Comparison of Different Methods by Simulations



Can we use logistic regression here?



$$\text{Log-likelihood: } l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ y_i(\mathbf{x}_i^T \boldsymbol{\beta}) - \log\left(1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta})\right) \right\}$$

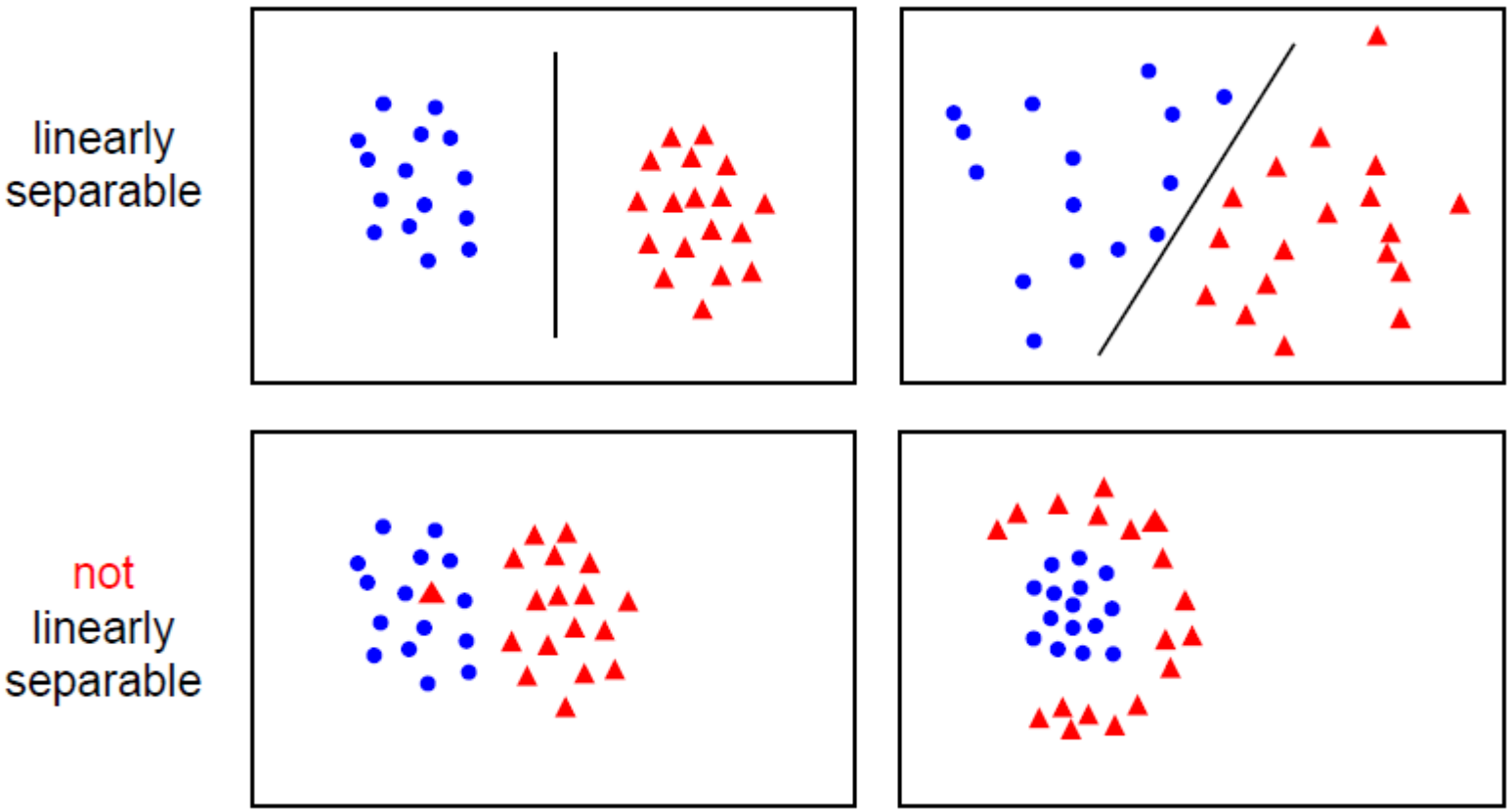
Separating Hyperplanes

- The methods considered so far use probabilistic arguments
- Now we will consider an approach that uses geometric arguments alone, without any probabilistic consideration
- Here we will consider linear decision boundaries (hyperplanes)

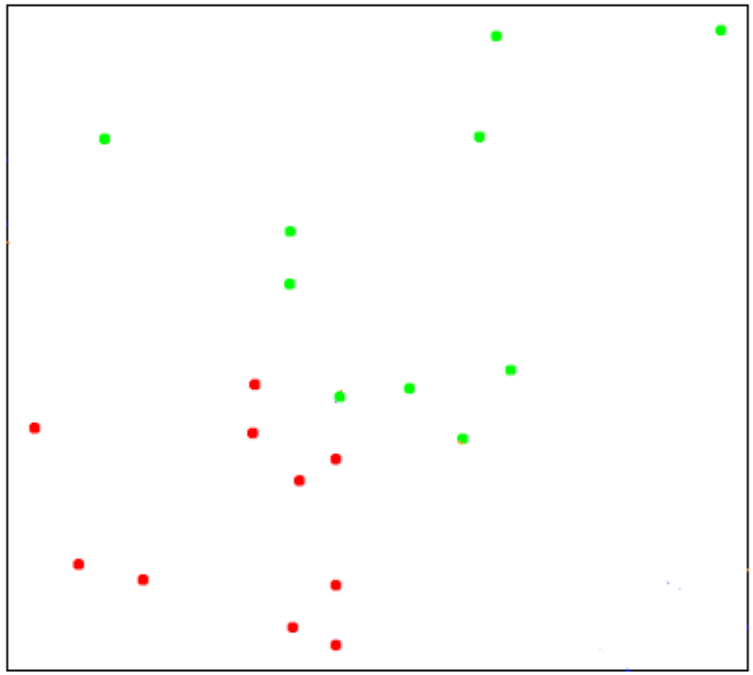
Linear separability

- Training data $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$.
- Goal: construct a classifier $G(\mathbf{x}) = \text{sign}[f(\mathbf{x})]$

Linear separability

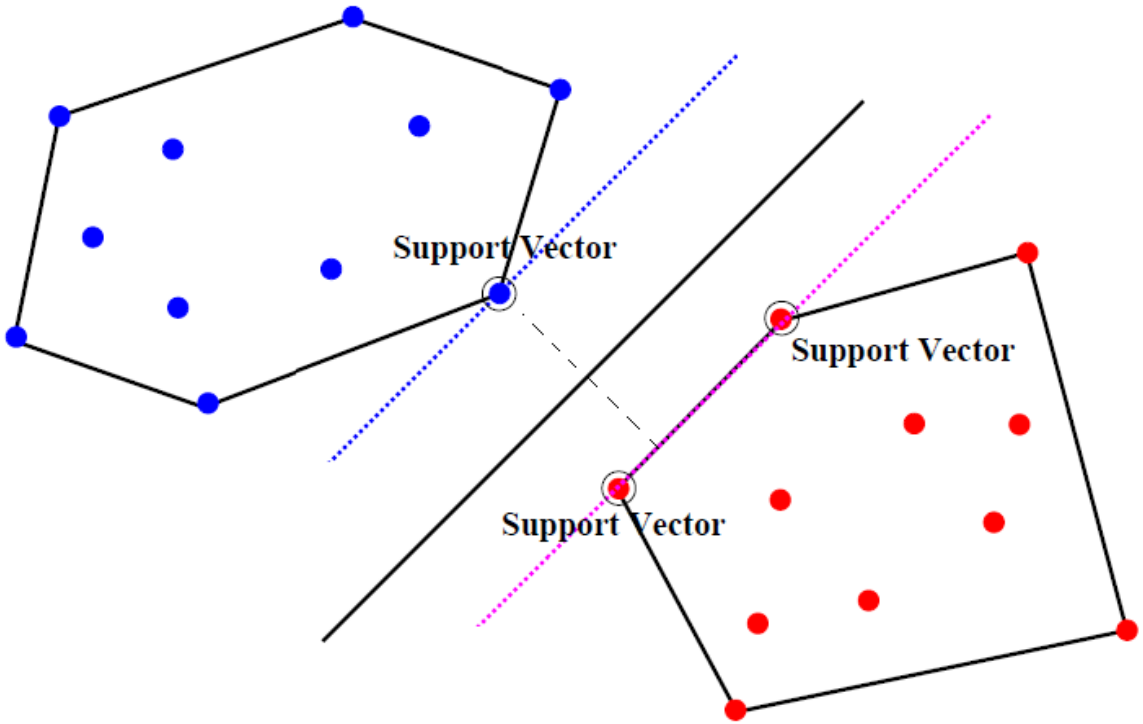


Separating Hyperplanes



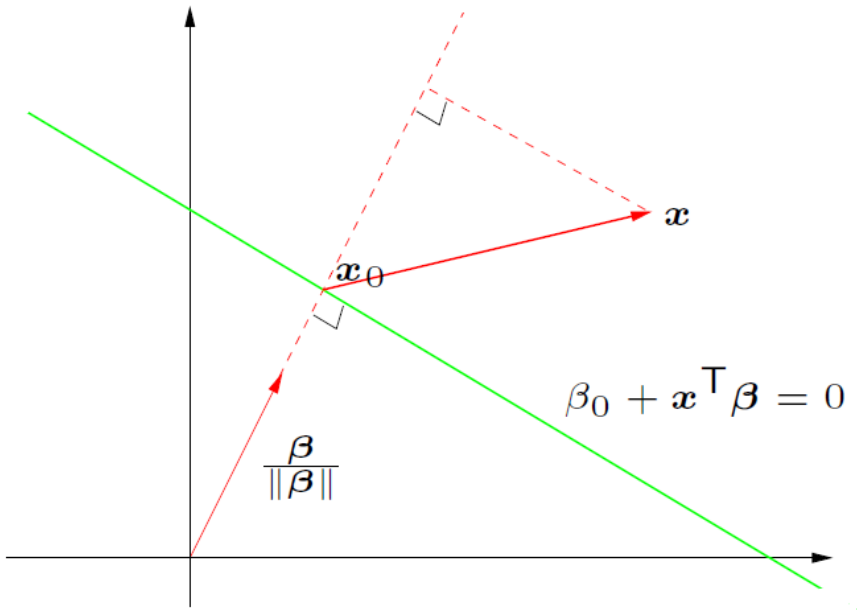
How to construct a linear classifier when the data are linear separable?

Geometric Algorithm



- Compute the convex hull of the positive points, and the convex hull of the negative points.
- Determine the shortest line between any two points along the convex hulls (one point on each convex hull).
- Draw the line that is perpendicular to the shortest line and yields the same margin on each side.

Signed Distance to Hyperplanes

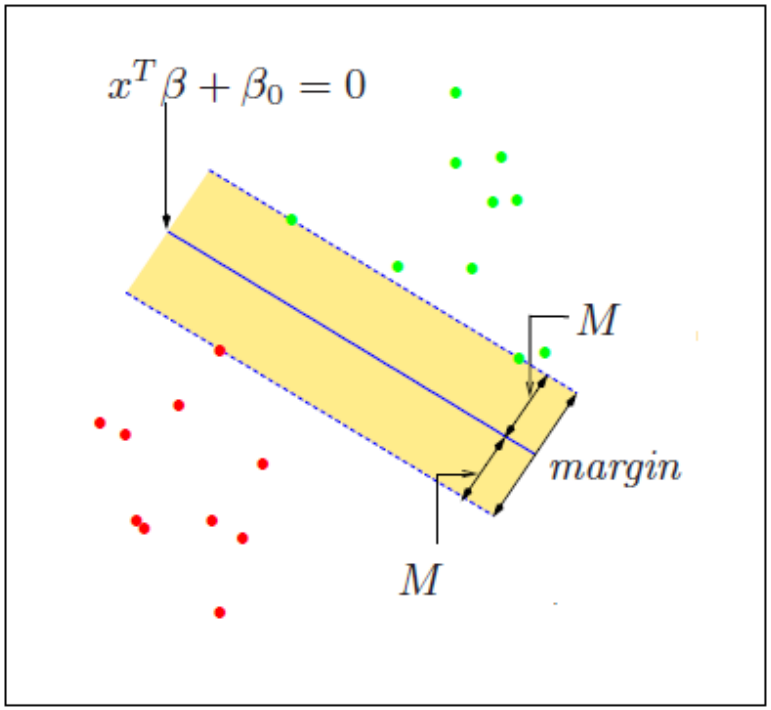


- Hyperplane is defined by $\{\mathbf{x} : f(\mathbf{x}) \triangleq \beta_0 + \mathbf{x}^T\boldsymbol{\beta} = 0\}$.
- Signed distance of point $\mathbf{x}$ to the plane is

$$\left\langle \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|}, \mathbf{x} - \mathbf{x}_0 \right\rangle = \frac{1}{\|\boldsymbol{\beta}\|}(\mathbf{x}^T\boldsymbol{\beta} - \mathbf{x}_0^T\boldsymbol{\beta}) = \frac{1}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}^T\boldsymbol{\beta})$$

Maximum Margin Classifier (Vapnik, 1995)

Assume the data can be separated by a linear boundary. $Y \in \{-1, 1\}$.



$$\max_{\beta_0, \boldsymbol{\beta}} M \qquad \text{s.t.} \qquad \frac{y_i}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq M, i = 1, \ldots, n.$$

- Maximize the minimum distance

Maximum Margin Classifier

- Let $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$ be the solution to

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 \qquad \text{s.t.} \qquad y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) \geq 1, i = 1, \ldots, n.$$

- The optimal separating hyperplane is

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}^T \hat{\boldsymbol{\beta}}.$$

- The maximal margin classifier is

$$\hat{G}(\mathbf{x}) = \text{sign}\big(\hat{f}(\mathbf{x})\big).$$

Lagrangian of Maximum Margin Classifier

Define the Lagrange primal

$$L_P(\beta_0, \boldsymbol{\beta}; \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{n} \lambda_i [1 - y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})]$$

where $\lambda_i \geq 0$, $i = 1, \ldots, n$, are Lagrange multipliers.

The minimizer satisfies (Karush-Kuhn-Tucker conditions)

- Stationarity:

$$\frac{\partial L_P(\beta_0, \boldsymbol{\beta}; \boldsymbol{\lambda})}{\partial \beta_0} = 0 \text{ and } \frac{\partial L_P(\beta_0, \boldsymbol{\beta}; \boldsymbol{\lambda})}{\partial \boldsymbol{\beta}} = 0.$$

- Primal feasibility: $y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) \geq 1, i = 1, \ldots, n.$
- Dual feasibility: $\lambda_i \geq 0, i = 1, \ldots, n.$
- Complementary slackness: $\lambda_i [y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) - 1] = 0, i = 1, \ldots, n.$

Lagrangian Primal and Dual

$$\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}) \quad \text{s.t. } g(\boldsymbol{\beta}) \leq 0 \text{ and } h(\boldsymbol{\beta}) = 0 \tag{1}$$

Lagrangian (Primal) $L_P(\boldsymbol{\beta}; \lambda, \mu) \triangleq Q(\boldsymbol{\beta}) + \lambda g(\boldsymbol{\beta}) + \mu h(\boldsymbol{\beta})$ with $\lambda \geq 0$.

The solution $\hat{\boldsymbol{\beta}}$ of (1) satisfies

$$Q(\hat{\boldsymbol{\beta}}) \geq L_P(\hat{\boldsymbol{\beta}}; \lambda, \mu) \geq \inf_{\boldsymbol{\beta}: g(\boldsymbol{\beta}) \leq 0, h(\boldsymbol{\beta})=0} L_P(\boldsymbol{\beta}; \lambda, \mu) \geq \inf_{\boldsymbol{\beta}} L_P(\boldsymbol{\beta}; \lambda, \mu) \triangleq L_D(\lambda, \mu)$$

for any $\lambda \geq 0$ and $\mu \in \mathbb{R}$. $L_D(\lambda, \mu)$ is called *Lagrange dual* function.

The Lagrange Dual Problem:

$$\max_{\lambda} L_D(\lambda, \mu) \quad \text{s.t.} \quad \lambda \geq 0.$$

- The dual solution $(\hat{\lambda}, \hat{\mu})$ satisfies weak duality: $L_D(\hat{\lambda}, \hat{\mu}) \leq Q(\hat{\boldsymbol{\beta}})$.
- The Lagrange dual problem is a convex optimization problem.

Strong Duality

$$\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}) \quad \text{s.t. } g(\boldsymbol{\beta}) \leq 0 \text{ and } h(\boldsymbol{\beta}) = 0 \tag{1}$$

Slater's Theorem: Suppose

- (1) is a convex optimization problem (i.e., $Q(\boldsymbol{\beta})$ and $g(\boldsymbol{\beta})$ are convex, $h(\boldsymbol{\beta})$ is linear).
- Slater's Condition: There exists a $\boldsymbol{\beta}$ s.t. $g(\boldsymbol{\beta}) < 0$ and $h(\boldsymbol{\beta}) = 0$.

Then strong duality holds: $L_D(\hat{\lambda}, \hat{\mu}) = Q(\hat{\boldsymbol{\beta}})$.

Under strong duality,

$$L_P(\hat{\boldsymbol{\beta}}; \hat{\lambda}, \hat{\mu}) = \inf_{\boldsymbol{\beta}: g(\boldsymbol{\beta}) \leq 0, h(\boldsymbol{\beta})=0} L_P(\boldsymbol{\beta}; \hat{\lambda}, \hat{\mu}) = \inf_{\boldsymbol{\beta}} L_P(\boldsymbol{\beta}; \hat{\lambda}, \hat{\mu}).$$

And thus,

- $\hat{\boldsymbol{\beta}} = \arg\inf_{\boldsymbol{\beta}: g(\boldsymbol{\beta}) \leq 0, h(\boldsymbol{\beta})=0} L_P(\boldsymbol{\beta}; \hat{\lambda}, \hat{\mu}) = \arg\inf_{\boldsymbol{\beta}} L_P(\boldsymbol{\beta}, \hat{\lambda})$.
- Primal feasibility: $g(\hat{\boldsymbol{\beta}}) \leq 0$ and $h(\hat{\boldsymbol{\beta}}) = 0$.
- Complementary slackness: $\hat{\lambda} g(\hat{\boldsymbol{\beta}}) = 0$.

Lagrange Dual for Maximum Margin Classifier

Lagrange primal function

$$L_P(\beta_0, \boldsymbol{\beta}; \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{n} \lambda_i [1 - y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})]$$

where $\lambda_i \geq 0$, $i = 1, \ldots, n$, are Lagrange multipliers.

Lagrange Dual Problem: $\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} L_D(\boldsymbol{\lambda})$ s.t. $\boldsymbol{\lambda} \geq 0$, where

$$L_D(\boldsymbol{\lambda}) = \min_{\beta_0, \boldsymbol{\beta}} L_P(\beta_0, \boldsymbol{\beta}; \boldsymbol{\lambda}) = \min_{\beta_0, \boldsymbol{\beta}} \left\{ \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{n} \lambda_i [1 - y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})] \right\}$$

$$= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k,$$

and λ_i's satisfy $\sum_{i=1}^{n} \lambda_i y_i = 0$.

Optimal Separating Hyperplane / Maximum margin classifier

The solution $\hat{\boldsymbol{\beta}}$ satisfies

- $(\beta_0, \boldsymbol{\beta}) = \arg\inf_{(\beta_0,\boldsymbol{\beta})} L_P(\beta_0, \boldsymbol{\beta}; \hat{\boldsymbol{\lambda}}) \quad \Longrightarrow \quad \hat{\boldsymbol{\beta}} = \sum_{i=1}^n \hat{\lambda}_i y_i \mathbf{x}_i..$
- Primal feasibility: $y_i(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\boldsymbol{\beta}}) \geq 1$ for all i.
- Complementary slackness: $\hat{\lambda}_i[y_i(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\boldsymbol{\beta}}) - 1] = 0$ for all i.

$\Rightarrow \hat{\beta}_0 = 1/y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}} = y_i - \sum_{k=1}^n \hat{\lambda}_k y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle$ for any i s.t. $\hat{\lambda}_i > 0$.

The optimal separating hyperplane is

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}^T \hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \sum_{i=1}^n \hat{\lambda}_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle.$$

The maximum margin classifier: $\hat{G}(\mathbf{x}) = \text{sign}\big(\hat{f}(\mathbf{x})\big)$.

Why Dual?

Let $\hat{\boldsymbol{\lambda}}$ be the solution to

$$\max_{\boldsymbol{\lambda}} \Big(\sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle \Big),$$

subject to $\sum_{i=1}^{n} \lambda_i y_i = 0$ and $\lambda_i \geq 0$ for $i = 1, \ldots, n$.

The maximum margin classifier $\hat{G}(\mathbf{x}) = \text{sign}\big(\hat{f}(\mathbf{x})\big)$, where

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\lambda}_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle.$$

- There are some quadratic programming algorithms that can solve the dual faster than the primal.
- $\hat{f}(\mathbf{x})$ only involve inner products of $\mathbf{x}_i$'s (kernel tricks ...).

More on Separating Hyperplane

- Sparse representation: the separating hyperplane $\hat{f}(\mathbf{x})$ is spanned by those data points i where $\hat{\lambda}_i > 0$, called *Support Vectors*.
- The Classifier $\hat{G}(\mathbf{x}) = \text{sign}\big(\hat{f}(\mathbf{x})\big)$ depends on the inner products of $\mathbf{x}$ with the support vectors.
- The optimal separating hyperplane $\hat{f}(\mathbf{x})$ is defined uniquely.

Example

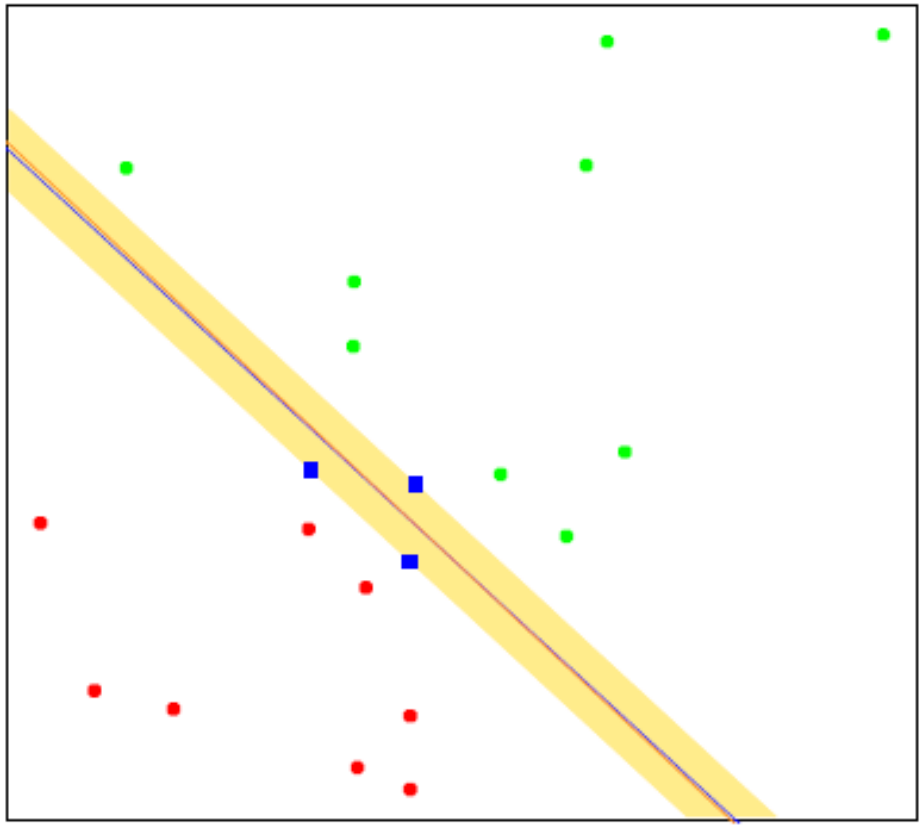
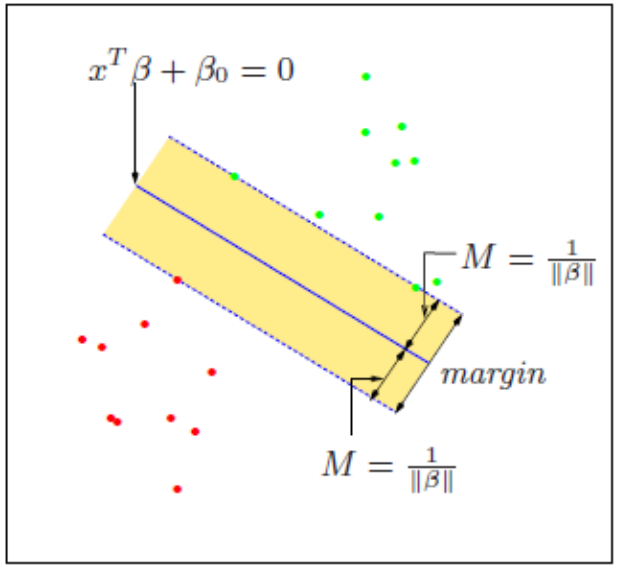
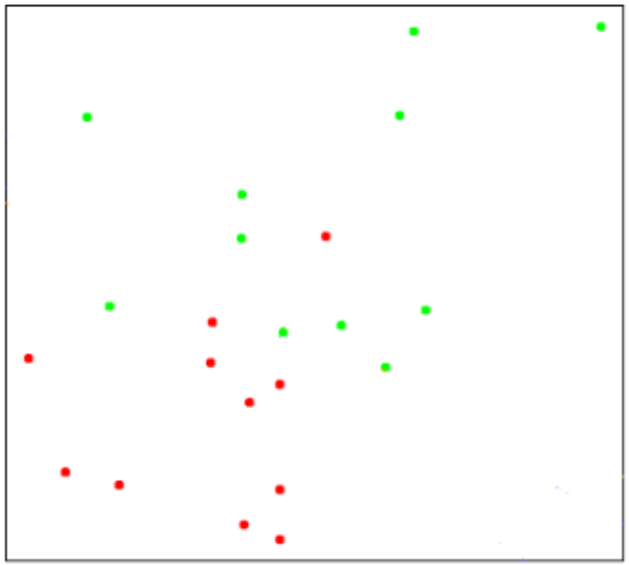


FIGURE 4.16. *The same data as in Figure 4.14. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).*

Overlapping Classes





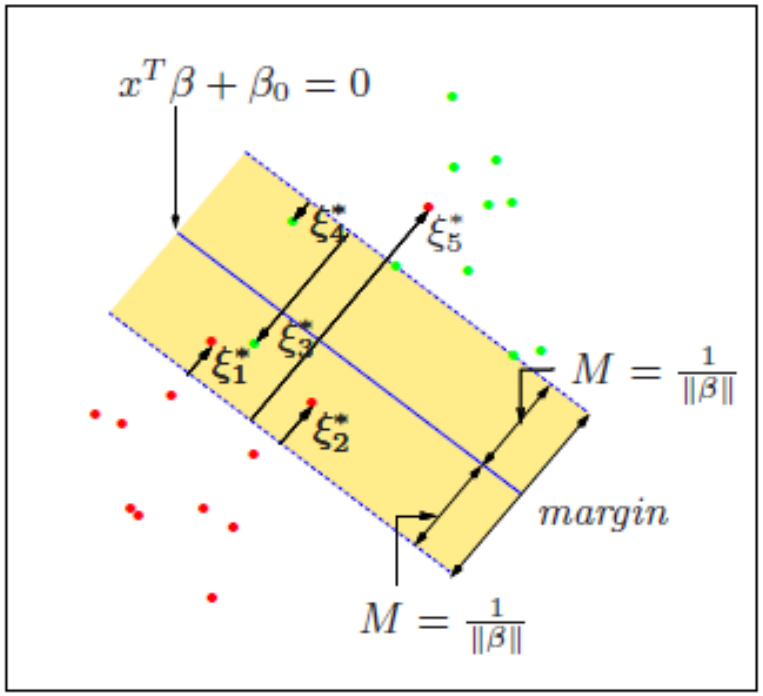
Linearly separable: $\max_{\beta_0,\boldsymbol{\beta}} M, \quad \text{s.t. } \frac{y_i}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq M, \forall i.$

Not linearly separable:

introduce slack variables ξ_i's: $\frac{y_i}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq M(1 - \xi_i), \xi_i \geq 0.$

An error occurs if $\xi_i > 1$.

Overlapping Classes: solutions



Solution:

$$\max_{\beta_0,\boldsymbol{\beta},\boldsymbol{\xi}} M,$$

$$\text{s.t. } \frac{y_i}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq M(1-\xi_i)$$

$$\xi_i \geq 0$$

$$\sum_i 1_{\xi_i > 1} \leq B.$$

where B is a tuning parameter.

Improved solution: replace $1_{\xi_i>1}$ by ξ_i (upper bound)

$$\max_{\beta_0,\boldsymbol{\beta},\boldsymbol{\xi}} M, \text{ s.t. } \frac{y_i}{\|\boldsymbol{\beta}\|}(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq M(1-\xi_i), \xi_i \geq 0, \sum_i \xi_i \leq B.$$

A small value of B will discourage any positive ξ_i.

Support Vector Machine

Setting $\|\boldsymbol{\beta}\| = 1/M$,

$$\min_{\beta_0, \boldsymbol{\beta}, \xi_i's} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n} \xi_i, \text{ s.t. } y_i(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq 1 - \xi_i, \xi_i \geq 0, \forall i,$$

where $C \geq 0$ is a tuning parameter (a large C discourages positive ξ_i).

The Lagrange primal is

$$L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \lambda_i[y_i(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) - (1 - \xi_i)] - \sum_{i=1}^{n} \mu_i\xi_i,$$

where $\lambda_i, \mu_i \geq 0$ are lagrange multipliers.

Lagrange Dual

Taking derivatives of L_P w.r.t. $\beta_0, \boldsymbol{\beta}$ and ξ_i's yields

$$\boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \quad 0 = \sum_{i=1}^{n} \lambda_i y_i, \text{ and } C = \lambda_i + \mu_i.$$

Substituting into the Lagrange primal, we obtain the Lagrange dual

$$L_D(\boldsymbol{\lambda}) = L_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle$$

The Lagrange dual problem:

$$\max_{\boldsymbol{\lambda}} L_D(\boldsymbol{\lambda}) \text{ s.t. } 0 \leq \lambda_i \leq C, \sum_{i=1}^{n} \lambda_i y_i = 0.$$

$$\Rightarrow \quad \hat{\boldsymbol{\lambda}}, \quad \hat{\mu}_i = C - \hat{\lambda}_i.$$

Support Vector Machine

The primal solution satisfies

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \hat{\lambda}_i y_i \mathbf{x}_i$$

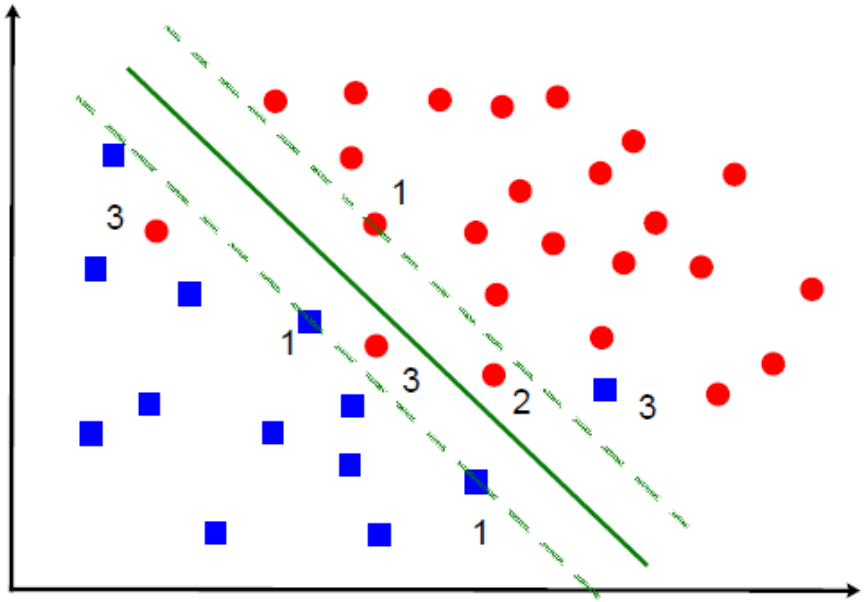
$$\hat{\lambda}_i [y_i(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\boldsymbol{\beta}}) - (1 - \hat{\xi}_i)] = 0, (C - \hat{\lambda}_i)\hat{\xi}_i = 0, \text{ for all } i.$$

$\Rightarrow \hat{\beta}_0 = 1/y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}} = y_i - \sum_{k=1}^{n} \hat{\lambda}_k y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle$ for any i s.t. $0 < \hat{\lambda}_i < C$.

The support vector classifier is

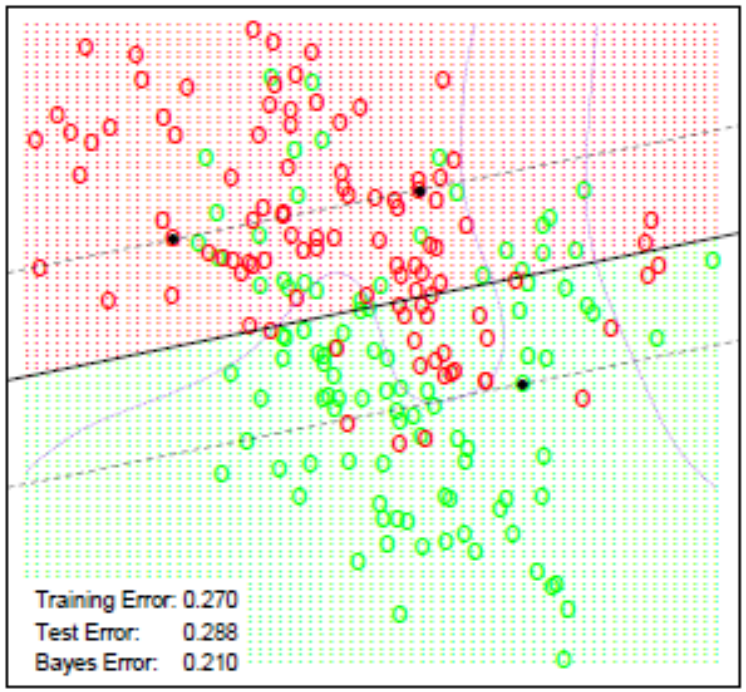
$$\hat{G}(\mathbf{x}) = \text{sign}(\hat{\beta}_0 + \mathbf{x}^T \hat{\boldsymbol{\beta}}) = \text{sign}\left(\hat{\beta}_0 + \sum_{i:\hat{\lambda}_i > 0} \hat{\lambda}_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle \right).$$

Support Vectors

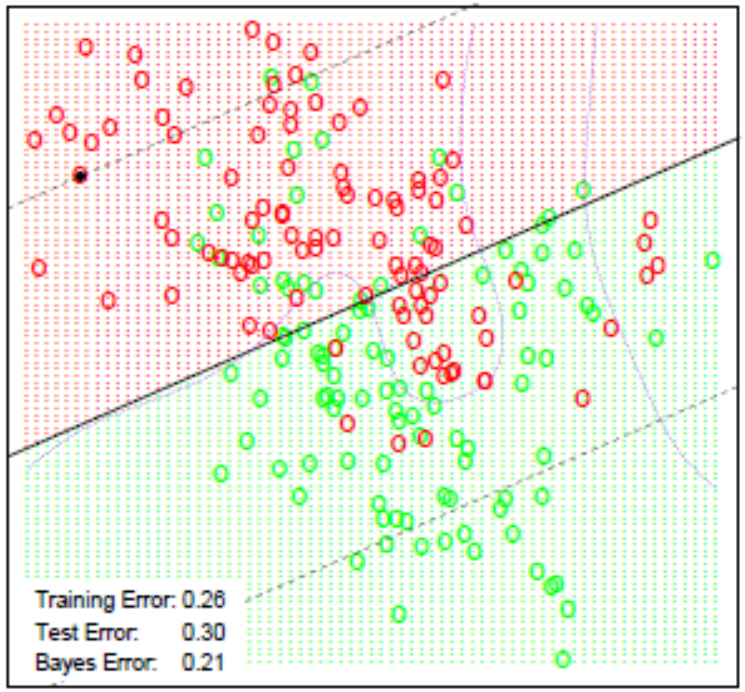


1. Margin vectors: $\hat{\xi}_i = 0,\ y_i(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}}) = 1$
2. non-margin vectors, correct specified:
$$0 < \hat{\xi}_i < 1,\ y_i(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}}) = 1 - \hat{\xi}_i.$$
3. non-margin vectors, misclassified:
$$\hat{\xi}_i > 1,\ y_i(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}}) = 1 - \hat{\xi}_i < 0.$$

Example: Linear SVM



$C = 10000$



$C = 0.01$

Flexible Classifiers

Enlarge the input space via basis expansion ($p \to q$):

$$\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_q(\mathbf{x})).$$

Lagrange dual and solution become

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \langle \mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_k) \rangle.$$

and

$$\hat{G}(\mathbf{x}) = \text{sign}\Big(\hat{\beta}_0 + \sum_{i:\hat{\lambda}_i > 0} \hat{\lambda}_i y_i \langle \mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}) \rangle \Big),$$

where $\hat{\beta}_0 = \text{ave}\left(y_i - \sum_{k=1}^{n} \hat{\lambda}_k y_k \langle \mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_k) \rangle \right)$ over all i s.t. $0 < \hat{\lambda}_i < C$.

Kernels

Both L_D and $\hat{G}(\mathbf{x})$ involve $\mathbf{h}(\mathbf{x})$ only through inner-products

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}') \rangle \qquad \text{– kernel}$$

Given a suitable kernel function $K(\mathbf{x}, \mathbf{x}')$, don't need $\mathbf{h}(\mathbf{x})$ at all.

Example: 2nd degree polynomial in $\mathbb{R}^2$:

If we choose $K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2$, then

$$K(\mathbf{x}, \mathbf{x}') = (1 + x_1 x_1' + x_2 x_2')^2 = \langle \mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}') \rangle,$$

where $\mathbf{h}(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$.

What's the computational cost of $(\mathbf{x}, \mathbf{x}') \to K(\mathbf{x}, \mathbf{x}')$ and $\mathbf{x} \to \mathbf{h}(\mathbf{x})$?

Popular Kernels

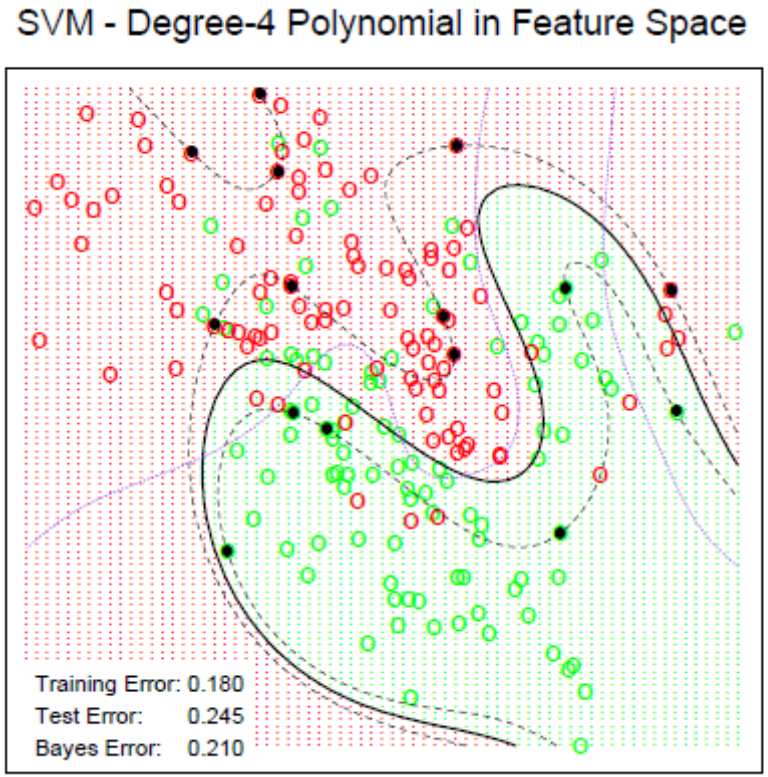
$K(\mathbf{x}, \mathbf{x}')$ is a symmetric, positive (semi-) definite function:

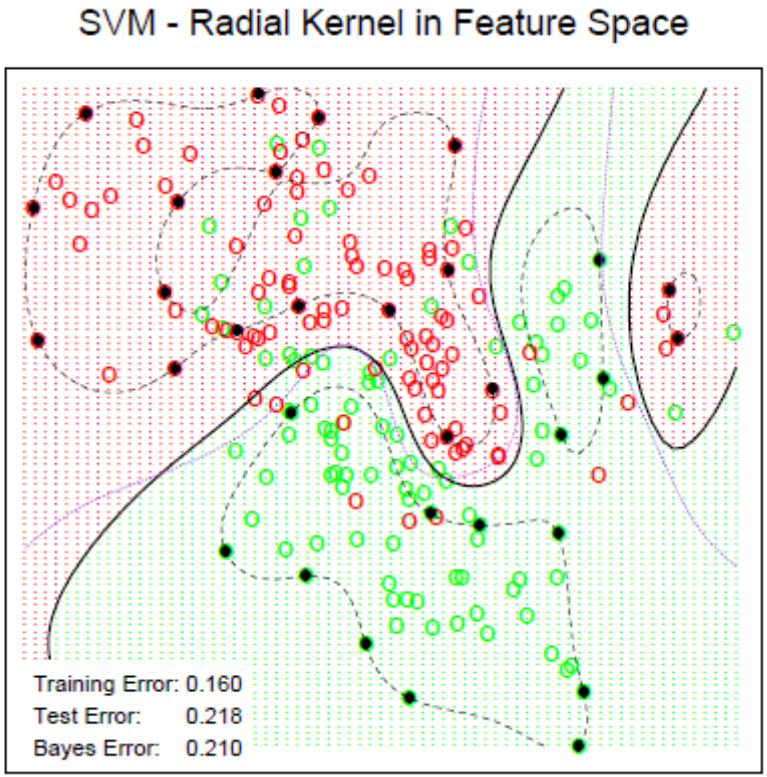
For every $n = 1, 2, \ldots$, and every set of real numbers $\{a_1, a_2, \ldots, a_n\}$ and $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, we have

$$\sum_{i=1}^{n} \sum_{k=1}^{n} a_i a_k K(\mathbf{x}_i, \mathbf{x}_k) \geq 0.$$

- dth-Degree polynomial: $K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$
- Radial basis $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2/\sigma^2\right)$.

Example continued: nonlinear SVMs





C was tuned to approx. achieve the best prediction performance, and $C = 1$.

SVM and Hinge Loss

The SVM solution

$$\min_{\beta_0,\boldsymbol{\beta},\xi_i's} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i, \text{ s.t. } y_i(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

is equivalent to the solution to

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

with $\lambda = 1/C$.

- Large $C \to$ small λ:

  overfit wiggly boundary if the function class is large.

- Small $C \to$ large λ: encourage small $\|\boldsymbol{\beta}\|$, smoother boundary.

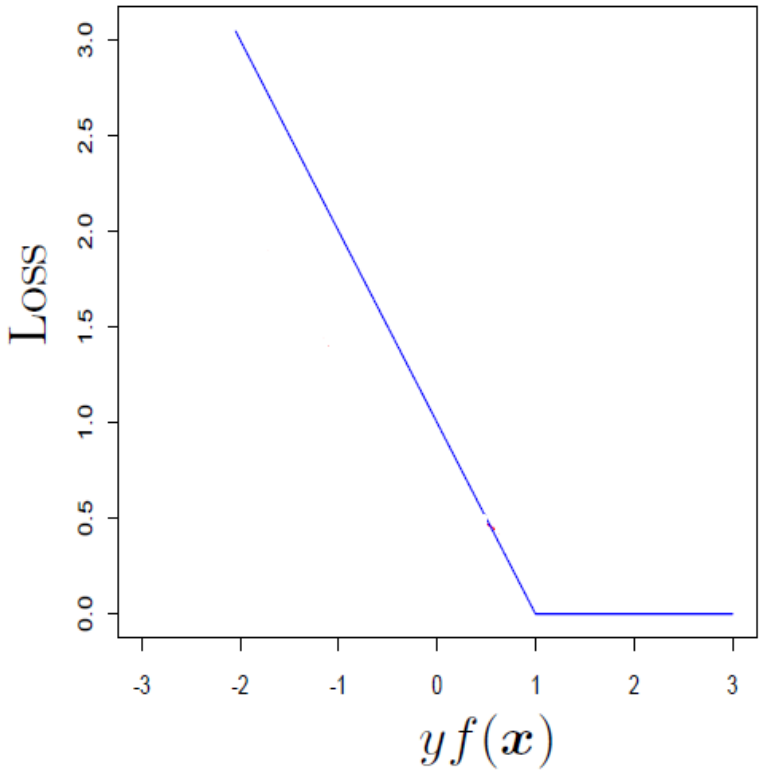
Hinge Loss

SVM: $L(y, f(\mathbf{x})) = (1 - yf(\mathbf{x}))_+$

- Called Hinge Loss.
- Estimate the classifier (threshold)

$$\text{sign}\big(P(Y = 1|\mathbf{x}) - 1/2\big)$$

- Implication on unequal cost



Group Paper Presentation

- V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, Nov. 7th, 2024

  Yangyang Chen, Serena Hu, Ze Li, Ziqiu Liu, Qu Sha, Eunice Wang

- Deep representation learning of electronic health records to unlock patient stratification at scale, Nov 14th, 2024

  Yimeng Cai, Yuxuan Du, Yuki Low, Hyunjee Oh, Haotian Tang, Yang Zhao

- U-Net: Convolutional Networks for Biomedical Image Segmentation, Nov 21st, 2024

  Manye Dong, Jiatong Li, Yiming Li, Wenxin Tian, Yuntian Xu, Shihang Zeng

- Deep-learning-based real-time prediction of acute kidney injury outperforms human predictive performance, Nov 21st, 2024

  Ruoying Deng, Ruijie He, Ekaterina Hofrenning, Jessie Li, Authur Starodynov, Yueyi Xu

- CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, Dec 5th, 2024

  Sixuan Chen, Aiying Huang, Yixiao Sun, Zihan Wu, Allison Xia, Jingyi Xu, Tongxi Yu

- Deep learning for the prediction of early on-treatment response in metastatic colorectal cancer from serial medical imaging, Dec 5th, 2024

  Mingzhi Chen, Chenshuo Pan, Zixuan Qiu, Xiaoting Tang, Mia Yu, Shubo Zhang

If you wish to swap presentation times, you may do so by mutually agreeing with a classmate. The deadline for any swaps is October 3rd.

Reminders

- Quiz is due on Monday (9/30) at 9pm.
- hw #1 is due on Saturday (10/5) at 9pm.