Logistic regression vs LDA

LDA
$$\log \frac{\Pr[G=0 \mid X=x]}{\Pr[G=1 \mid X=x]} = \log \frac{\pi_0}{\pi_1} \underbrace{- \frac{1}{2}(\mu_0+\mu_1)^T \Sigma^{-1}(\mu_0-\mu_1)}_{\alpha_{01}} + x^T \underbrace{\Sigma^{-1}(\mu_0-\mu_1)}_{\alpha_1}$$

$$= \alpha_{01} + \alpha_1^T x$$

similer to the logistic regression

$$= \beta_{01} + \beta_1^T x$$

$$\Pr[G=k, X=x] = \Pr(x) \underbrace{\Pr[G=k \mid X=x]}_{\text{both LDA and logistic regress.}}$$

$$\frac{e^{\beta_{0k} + \beta_k^T x}}{1 + \sum_{\ell=1}^{K} e^{\beta_{0\ell} + \beta_\ell^T x}}$$

LDA also model this part

$$\Pr(x) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \Sigma)$$

gaussian

Model assessment and model selection

① ②

① evaluate the performance (in term of prediction) of a selected model

② select the best models (for prediction)

GOAL of a prediction model
generalization : a prediction model must be valid in broad generality, and not valid for a specific dataset

Bias, variance, model complexity

Define:

Y = target variable

X = input matrix

$\hat{f}(x)$ prediction rule, that is trained on a training set τ

Error is measured through a loss function

$$L(Y, \hat{f}(x))$$

that should penalize differences between Y and $\hat{f}(x)$

typical choice for continuous outcomes

$$L(Y, \hat{f}(x)) = \begin{cases} (Y - \hat{f}(x))^2 & \text{quadratic loss} \\ |Y - \hat{f}(x)| & \text{absolute loss} \end{cases}$$

Test error (generalization error) is a prediction error computed on an <u>independent</u> sample

$$Err_{\tau} = E[L(Y, \hat{f}(x)) \mid \tau]$$

random, X, Y

τ is fixed, is the specific training sets on which we derive our prediction rule

In general, we would like to minimize the expected prediction error

$$Err = E[L(Y, \hat{f}(x))] = E[Err_{\tau}]$$

We do not want a model with the smallest prediction error for the specific training set, but for the general case

Since we have our training set, we are going to estimate $Err_{\mathcal{T}}$

$\hookrightarrow$ $\underbrace{\text{the training}}$ error is <u>NOT</u> a good estimate of $Err_{\mathcal{T}}$

$$\hookrightarrow \overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i; f(x_i))$$

We do <u>NOT</u> want to minimize the training error.
We saw that by increasing the model complexity, we can always decrease the training error

OVERFITTING → our model is specific for the training set
→ generalizes poorly

Similar story for categorical outcome

indicator function
$\mathbb{1}(x \neq x_0) \begin{cases} 1 & x \neq x_0 \\ 0 & x = x_0 \end{cases}$

G : target variable → takes K values in $\mathcal{G}$

typical loss functions in this case

$$L(G, \hat{G}(x)) \cdot \begin{cases} \underline{\mathbb{1}(G \neq \hat{G}(x))} & \text{0-1 loss} \\ -2 \underbrace{\log L(\beta)}_{\ell(\beta)} & \text{deviance} \end{cases}$$

- $-2\,\ell(\beta)$ is a general loss-function, it can be used in all cases (binomial, Gamma, Poisson, log-normal, ...)

- the factor -2 is added to make the loss function be equal to the squared loss in the Gaussian case

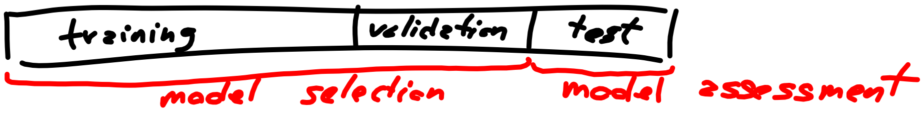
$$L(\beta) = \frac{1}{\sqrt{2\pi 1}} \exp\left\{-\frac{1}{2\cdot 1}\sum_{i=1}^{N}(y_i - x_i^T\beta)^2\right\}$$

$$\ell(\beta) = -\frac{1}{2}\underbrace{\sum_{i=1}^{N}(y_i - x_i^T\beta)^2}_{\text{sum of squares}}$$

$-2\,\ell(\beta) =$ squared loss

In an ideal situation
↳ a lot of data

we can randomly split the observations in three independent sets



- training set : contains the data on which we fit our models
- validation set : data we use to identify the best model
- test set : data to assess the performance of the selected model

↳ this set must be considered only at the end of the analysis
i.e. only when we have already chosen the best model
Must be ignored for model selection
– avoid overoptimism

How to split the data in the three sets :

- no general rule
- book suggestion is :

50%	training set
25%	validation set
25%	test set

- it depends:
  + sample size
  + on the signal-to-noise ratio
  + complexity of the model we are considering

The Bias-Variance Decomposition

$Y = f(x) + \varepsilon$, $E[\varepsilon] = 0$
$Var[\varepsilon] = \sigma_\varepsilon^2$

$$Err(x_0) = E[(Y - \hat{f}(x_0))^2 \mid X = x_0]$$

$$= \sigma_\varepsilon^2 + (E[\hat{f}(x)] - f(x_0))^2 + E[(\hat{f}(x_0) - E[\hat{f}(x_0)])^2]$$

$=$ irreducible error + bias^2 + variance

where

- σ_ε^2 , the variance of the target around the true mean, so we cannot do anything about that
- bias^2, the squared difference between the average of our estimates and the true mean
- variance, the expected squared difference between $\hat{f}(x)$ and its mean

KNN

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_k(x_0)} y_i$$

$$Err(x_0) = \sigma_\varepsilon^2 + \left[f(x_0) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})\right]^2 + \frac{\sigma_\varepsilon^2}{K}$$

$$E[\hat{f}(x)] = E\left[\frac{1}{K}\sum_{\ell=1}^{K} Y_{(\ell)}\right] = \frac{1}{K}\sum_{\ell=1}^{K} E[Y_{(\ell)}] = \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})$$

$$Var[\hat{f}(x_0)] = Var\left[\frac{1}{K}\sum_{\ell=1}^{K} Y_{(\ell)}\right] = \frac{1}{K^2}\sum_{\ell=1}^{K} Var[Y_{(\ell)}] = \frac{1}{K^2}\sum_{\ell=1}^{K}\sigma_\varepsilon^2 = \frac{\cancel{K}}{K^2}\sigma_\varepsilon^2 = \frac{\sigma_\varepsilon^2}{K}$$

$$\#K \propto \frac{1}{\text{complexity}}$$

increase K → decrease complexity
→ more bias
→ reduce the variance

similar for linear model $\hat{f}(x;\beta) = x^T\beta$

- slightly more difficult to derive, it turns out to be

$$\frac{1}{N}\sum_{i=1}^{N} Err(x_i) = \sigma_\varepsilon^2 + \underline{\frac{1}{N}\sum_{i=1}^{N}\left[f(x_i) - E[\hat{f}(x_i)]\right]^2} + \frac{p}{N}\sigma_\varepsilon^2$$

p ↗

called the in-sample error

p = # of variables

increasing the model complexity, we increase the variance component of the error

For regularized regression (e.g., lasso, ridge, ...) the form is the same, but there is an additional dependence on the tuning (complexity) parameter α

- we can go more into details on the bias component

$$E_{x_0}\left[f(x_0) - E[\hat{f}(x_0)]\right]^2 = E_{x_0}\left[f(x_0) - x^T\hat{\beta}_*\right] + E_{x_0}\left[x_0^T\beta - E[x_0^T\hat{\beta}_\alpha]\right]^2$$

$$\hat{\beta}_* = \arg\min_\beta E[f(x) - x^T\beta)^2$$

Average [model bias]^2 + Ave [Estimation bias]^2

difference between the true function and the best fitting linear approximation

error between the average estimate and the best model

difference between truth and best model

- we can reduce it only increasing the model space (more variables, more complex relationship - interaction -, ...)

additional bias that we add when, e.g., add shrinkage

OLS = 0