Recall

1. Logistic Regression. (glm).
2. Introduced how to evaluate a classifier.

Traning - Testing Split:

data $D = \{(x_i, y_i)\}_{i=1}^{n}$.

Let $\mathcal{A}$ be an algorithm s.t. $\mathcal{A}: P(D) \to \{g: \mathbb{R}^p \to \{0,1\}\}$.

$P(D) = \{V : V \subseteq D\}$. set of all subsets

$\{g: \mathbb{R}^p \to \{0,1\}\}$. set of all classifiers.

Given data D, we can train a classifier $g = \mathcal{A}(D)$.

$D = D_{train} \cup D_{test}$.

$g = \mathcal{A}(D_{train})$ is our classifier.

def. $err_{train} = \frac{1}{|D_{train}|} \sum_{i: x_i \in D_{train}} I(y_i \neq g(x_i))$.

def $err_{test} = \frac{1}{|D_{test}|} \sum_{i: x_i \in D_{test}} I(y_i \neq g(x_i))$.

We argued that we cannot evaluate how "good" a classifier is solely based on Training error.

If we take $g(x_i) = y_i$. then, $err_{train} = 0$.

It may do a good job on training set. but it cannot generalize to testing set.

If $err_{test} >> err_{train}$ $\Rightarrow$ overfitting.

$err_{test} << err_{train}$ $\Rightarrow$ underfitting.

Drawback: We loss some efficiency.

K-fold cross-validation. (K-fold CV)

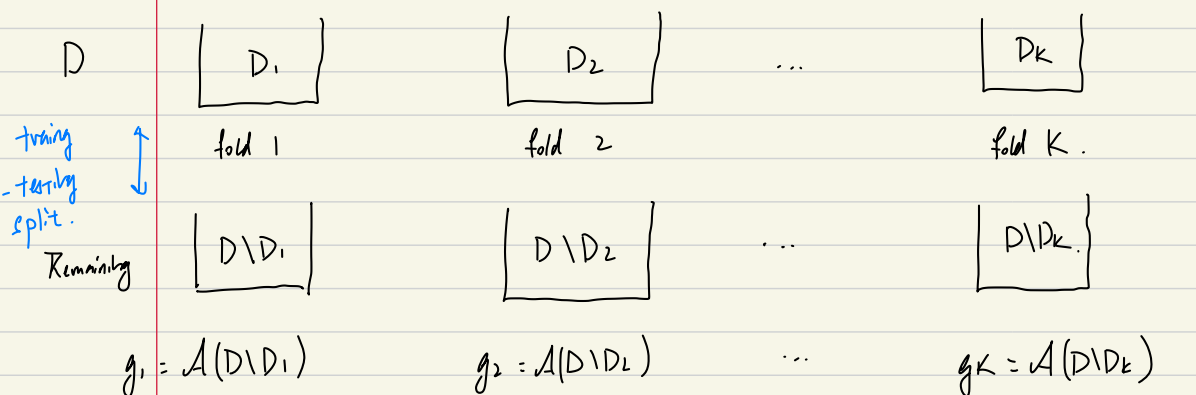
$D = \bigcup_{i=1}^{K} D_i$ each D_i is a "sub-test" set

$g_i = A(D \setminus D_i)$

$err_{test}^{(i)} = \sum_{j : x_j \in D_j} I(y_j \neq g_i(x_j))$.

CV-score $= \sum_{j=1}^{K} err_{test}^{(i)}$.

(total error)



$g_1 = A(D \setminus D_1)$ $g_2 = A(D \setminus D_2)$ ... $g_K = A(D \setminus D_K)$

$err_{test}^{(i)}$ $\sum_{j : x_j \in D_1} I(y_j \neq g_1(x_j))$ $\sum_{j : x_j \in D_2} I(y_j \neq g_2(x_j))$ ... $\sum_{j : x_j \in D_K} I(y_j \neq g_K(x_j))$

$$CV\text{-score} = \sum_{i=1}^{K} \sum_{j : x_j \in D_i} I(y_j \neq g_i(x_j)).$$

Special case of CV: Leave-One-Out CV (LOOCV).
LOOCV is n-fold CV.

thm. For Linear Regression. LOOCV score is given by

$$\text{CV-score} = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2 . \qquad h_{ii} = \left[X (X^T X)^{-1} X^T \right]_{ii}$$

$$\hat{y} = X\hat{\beta} = \underbrace{X (X^T X)^{-1} X^T}_{H} y = Hy . \qquad H = \begin{bmatrix} h_{11} & & * \\ & h_{22} & \\ * & & \ddots \\ & & & h_{nn} \end{bmatrix}$$

proof.

i-th fold, Test set: $\{x_i, y_i\} = D_i$ Training set $\{(x_j, x_j)_{j \neq i}\} = D \setminus D_i$

$$\tilde{y}_i = g(x_i), \quad g = A(D \setminus D_i)$$

$$= x_i^T \left(X_{(-i)}^T X_{(-i)} \right)^{-1} X_{(-i)}^T y_{(-i)} .$$

$$D \setminus D_i = (X_{(-i)}, y_{(-i)}), \quad X_{(-i)} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{i-1} \\ 1 & x_{i+1} \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \qquad y_{(-i)} = \begin{pmatrix} y_1 \\ \vdots \\ y_{i-1} \\ y_{i+1} \\ \vdots \\ y_n \end{pmatrix}.$$

thm, it is clear $\hat{\beta}_{(-i)} = \left(X_{(-i)}^T X_{(-i)} \right)^{-1} X_{(-i)}^T y_{(-i)}$ $\left((X^T X)^{-1} X^T y \right)$.

hence, $\tilde{y}_i = x_i^T \hat{\beta}_{(-i)} = x_i^T \left(X_{(-i)}^T X_{(-i)} \right)^{-1} X_{(-i)}^T y_{(-i)}$.

$$X_{(-i)}^T X_{(-i)} = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_j x_j^T = \sum_{j=1}^{n} x_j x_j^T - x_i x_i^T = X^T X - x_i x_i^T .$$

$$X_{(-i)}^T y_{(-i)} = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_j y_j = \sum_{j=1}^{n} x_j y_j - x_i y_i .$$

$$(X^TX - x_i x_i^T)^{-1} = (X^TX)^{-1} + \frac{(X^TX)^{-1} x_i x_i^T (X^TX)^{-1}}{1 - x_i^T (X^TX)^{-1} x_i}.$$ Sherman-Morrison.

$$X_{(-i)}^T y_{(-i)} = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_j y_j = \sum_{j=1}^{n} x_j y_j - x_i y_i = X^T y - x_i y_i.$$

$$\tilde{y}_i = x_i^T \hat{\beta}_{(-i)} = x_i^T (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T y_{(-i)}.$$

$$= x_i^T (X^TX)^{-1} X^T y - x_i^T (X^TX)^{-1} x_i y_i.$$

$$+ \frac{x_i^T (X^TX)^{-1} x_i x_i^T (X^TX)^{-1} X^T y}{1 - x_i^T (X^TX)^{-1} x_i} - \frac{x_i^T (X^TX)^{-1} x_i x_i^T (X^TX)^{-1} x_i y_i}{1 - x_i^T (X^TX)^{-1} x_i}.$$

Note: $X (X^TX)^{-1} X^T = H$.
$x_i^T (X^TX)^{-1} x_i = h_{ii}$.
$x_i^T (X^TX)^{-1} X^T y = \hat{y}_i$.

$$= \hat{y}_i - h_{ii} y_i + \frac{h_{ii} \hat{y}_i}{1 - h_{ii}} - \frac{h_{ii}^2 y_i}{1 - h_{ii}}.$$

$$= \hat{y}_i \left\{ 1 + \frac{h_{ii}}{1 - h_{ii}} \right\} - h_{ii} y_i \left\{ 1 + \frac{h_{ii}}{1 - h_{ii}} \right\}.$$

$$\tilde{y}_i = \hat{y}_i \frac{1}{1 - h_{ii}} - h_{ii} y_i \frac{1}{1 - h_{ii}} = \frac{1}{1 - h_{ii}} (\hat{y}_i - h_{ii} y_i).$$

$\tilde{y}_i$ = predicted from $X_{(-i)}$, $y_{(-i)}$.
$\hat{y}_i$ = predicted from X, y.

$$y_i - \tilde{y}_i = y_i - \frac{\hat{y}_i}{1 - h_{ii}} + \frac{h_{ii}}{1 - h_{ii}} y_i = \frac{1}{1 - h_{ii}} y_i - \frac{1}{1 - h_{ii}} \hat{y}_i = \frac{y_i - \hat{y}_i}{1 - h_{ii}}.$$

$$\Rightarrow \text{CV-score} = \sum_{j=1}^{n} (y_i - \tilde{y}_i)^2 = \sum_{j=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2.$$

thm. If $\hat{y} = Sy$, where S is a matrix, thm, (S linear smoother)

$$LV\text{-score} = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2$$

if $h_{ii} \approx 1$, thm, this not numerically stable.

$$\text{Generalize CV (GCV)} = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{(1 - \text{tr}(S)/n)^2} \qquad \text{tr}(S) = \sum_{i=1}^{n} h_{ii}$$

Unfortunately, logistic regression is not a linear smoother.

Smoothing Technique. (Spline). } ⇒ CV closed form available.
Kernel Trick (KDE). }

Use of CV.

1. To evaluate the performance of classifier.
2. Hyperparameter selection.

HW2 Q2. Regularized Logistic Regression.

Logistic Regression: $l(\beta) = \sum_{i=1}^{n} \text{loglik}(\beta)$.

Regularize Logistic Regression. $pl(\beta, \lambda) = \sum_{i=1}^{n} \text{loglik}(\beta) + \lambda \|\beta\|_2^2$ (λ: hyperparameter.)

↓

Ridge Regression. ⇒ $l(\beta) + \lambda \|\beta\|_2^2$ $\quad \|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$

LASSO Regression. ⇒ $l(\beta) + \lambda \|\beta\|_1$ $\quad \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$

$$CV(\lambda) = \sum_{j=1}^{K} \sum_{i : x_i \in D_j} I\left(y_i \neq g_j(x_i, \lambda)\right)$$

$$\hat{\lambda} = \underset{\lambda}{\text{argmin}}\ CV(\lambda)$$