FoDA - Lib3

Cross - Validation
How well is regression working?

Input \((x_1, y)\), \(y \in \mathbb{R}^n\)

convert \(x \rightarrow \hat{x}_p = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_p \end{bmatrix} \)

\[
\alpha^*_p = (\hat{x}_p^T \hat{x}_p)^{-1} \hat{x}_p^T y
\]

\(\alpha_p = (\alpha_0, \ldots, \alpha_p) \in \mathbb{R}^{p+1}\)

\[
M_{\alpha_p}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_p x^p
\]

Which choice of \(p\)?
$y_i = l(x_i) + N(0, \sigma^2)$

$\text{How well does generalizer } M \text{ to new data?}$

$\text{SSE} \left( l(x_i), M_{\alpha^{*}} \right) = 0$
What makes a model good?

- Stable to noise ($x$-coord)
- Fit data well (y noise tolerance)
- Simple as possible

How well does it generalize to new data?
Cross-Validation

Assume \((X,y)\) iid \(\neq \)

New data \((x', y')\) iid \(\neq \)

Assume \(n\) (number of data points) is large

Randomly split \((X,y)\) into \((X_{\text{train}}, Y_{\text{train}})\)

and a test set \((X_{\text{test}}, Y_{\text{test}})\)
Usually:

- 70% in train
- 90% in train
- 30% in test
- 10% in test

Procedure:

1. Solve
2. Train
3. Evaluate

Train data:
\[ \hat{\beta}_{\text{train}} = (X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T y_{\text{train}} \]

Evaluate:
\[ \text{SSE}(\hat{\beta}) = \sum_{(x_i, y_i) \in X_{\text{test}}} (y_i - f_{\hat{\beta}}(x_i))^2 \]

Where: $\text{SSE}$ is the sum of squared errors, $f_{\hat{\beta}}(x_i)$ is the predicted value for $x_i$, and $y_i$ is the actual value.

Note: New data looks like this.
What is cross-validation good for?

1. Predict how well does $M_a$ do on new data.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y - M_a(x))^2}$$

2. Choose the best model (parameter $e.g. P$)

$MSE((x_u, 1998), M_{X, 2})$ vs. $MSE((x_u, Y_t), M_{a, 7})$
\[ p^* = \text{arg min}_p \quad \text{SSE}(\mathbf{x}_{te}, \mathbf{y}_{te}, \mathbf{M}_p) \]

\[ = \text{arg min}_p \quad \text{RMSE}(\mathbf{x}_{te}, \mathbf{y}_{te}, \mathbf{M}_p) \]

\[ = \sqrt{\frac{1}{n_{te}} \text{SSE}} \]
What to do if not enough data to split?

---

Leave-one-out C-V in data points

Create $n$ test / train splits

$X_{tr, i} \forall i \in [1, \ldots, n]$

$X_{te, i} = \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$

$X_{te, i} = \{x_i\}$
Error = \sum_{i=1}^{\infty} \left( y_0^i - M_{x_{test}}(x_0) \right)^2

Single test point for each model split.

Build model $M_{x_{test}}$ for each split.