

8.2 LOOCV

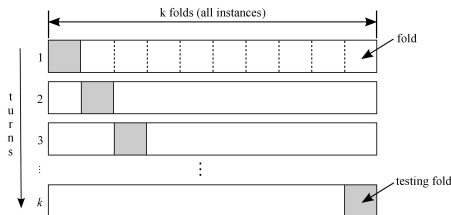
Machine Learning 1: Foundations

Marius Kloft (TUK)

- 1 Linear Regression
- 2 LOOCV
- 3 Non-linear Regression
 - Kernel Ridge Regression
 - Deep Regression
- 4 Unifying Loss View of Regression and Classification

How to Select the Regularization Parameter C ?

Use k -fold cross validation (CV), introduced in lecture 1:



- 1: split data into k ^{e.g.} 10 equally-sized chunks (called “folds”)
- 2: **for** $i = 1, \dots, k$ and $C \in \{0.01, 0.1, 1, 10, 100\}$ **do**
- 3: use i th fold as **test set** and union of all others as **training set**
- 4: train learner on training set (using C) and test on test set
- 5: **end for**
- 6: output learner with lowest average error

Similarly, can select constants in other learning methods, e.g.:

- RBF-kernel width in SVM, learning rate in ANNs, etc.

But What Does **Error** Mean in Regression?

In binary classification, we had $y \in \{-1, +1\}$

- ▶ so we could just count the fraction of correctly classified test instances (the **accuracy**)

In regression, y can attain any value: $y \in \mathbb{R}$

- ▶ whether the prediction is right or wrong is not the point here
- ▶ the point is by how much the prediction is wrong

The common error measure in regression is:

Definition

Let $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ be a test set, and let f be a learned regression function. The **root mean squared error (RMSE)** of f is:

$$\text{RMSE}(f) := \sqrt{\frac{1}{n} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2}$$

Sometimes We Have Very Little Data



Predicting the effect of an anti-cancer drug on tumors in mice

- ▶ typically $n \ll 100$

How can we use **as much data as possible**
in cross-validation?

Leave ONE Point Out For Testing and Use ALL Others For Training:

Definition

Leave-one-out cross-validation (LOOCV) is k -fold CV

- ▶ with $k := n$

In other words:

- ▶ we have as many folds as data points
- ▶ each fold contains only a single point

Theoretically, LOOCV is the best procedure to select constants, such as C

But what could be a problem with LOOCV?

LOOCV is Usually Super Slow

Involves a loop over all data points: $O(n)$

- ▶ In each iteration, train learner with $n - 1$ data points:
 - ▶ is $O(d^3)$ for RR

Total LOOCV (for RR): $O(d^3 n)$

Can we get rid of the loop over all data points?

- ▶ for ridge regression: yes!
- ▶ for classification: no!

LOOCV Trick for RR

The LOOCV error is:

$$\text{RMSE}_{\text{loocv}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{w}_i - y_i)^2}$$

where:

- ▶ \mathbf{w}_i is RR solution when i th data point is left out at training

$$\text{Recall: } \mathbf{w}_{\text{RR}} = \left(\underbrace{XX^\top}_{=\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top} + \frac{1}{2C} I \right)^{-1} \underbrace{Xy}_{=\sum_{i=1}^n \mathbf{x}_i y_i}$$

$$\text{Thus: } \mathbf{w}_i = (XX^\top - \mathbf{x}_i \mathbf{x}_i^\top + \frac{1}{2C} I)^{-1} (Xy - \mathbf{x}_i y_i)$$

Problem:

- ▶ Need to invert the matrix occurring in \mathbf{w}_i for all $i = 1, \dots, n$
- ▶ Each inversion is $O(d^3) \Rightarrow$ total: $O(d^3 n)$

Turns out: ONE matrix inversion suffices (total of $O(d^3)$).

How does this trick work?

Skipping the Matrix Inversion—Here's the Trick:

Write:
$$\mathbf{w}_i = \left(\underbrace{XX^\top + \frac{1}{2C}I}_{=:A} - \underbrace{\mathbf{x}_i\mathbf{x}_i^\top}_{\mathbf{u}\mathbf{u}^\top} \right)^{-1} (Xy - \mathbf{x}_i y_i)$$

Apply the following theorem:

Theorem (Sherman-Morrison formula)

Let $A \in \mathbb{R}^{d \times d}$ be an invertible matrix, and let $\mathbf{u} \in \mathbb{R}^d$.
If $\mathbf{u}^\top A^{-1} \mathbf{u} \neq 1$, then:

$$(A - \mathbf{u}\mathbf{u}^\top)^{-1} = A^{-1} + \frac{A^{-1}\mathbf{u}\mathbf{u}^\top A^{-1}}{1 - \mathbf{u}^\top A^{-1} \mathbf{u}}$$

Thus:
$$\text{RMSE}_{\text{loocv}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{w}_i - y_i)^2}$$

$$= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^\top \left(A^{-1} + \frac{A^{-1} \mathbf{x}_i \mathbf{x}_i^\top A^{-1}}{1 - \mathbf{x}_i^\top A^{-1} \mathbf{x}_i} \right) (Xy - \mathbf{x}_i y_i) - y_i \right)^2}$$

The Last Equation From the Previous Slide ...

... shows already that we can come along with a total of $O(d^3)$ to compute the LOOCV error.

But we can further simplify the expression and obtain:

Theorem

The LOOCV-RMSE of ridge regression can be computed in $O(d^3)$ through:

$$\text{RMSE}_{\text{loocv}} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\mathbf{x}_i^\top \mathbf{w}_{\text{RR}} - y_i}{1 - \mathbf{x}_i^\top \mathbf{A}^{-1} \mathbf{x}_i} \right)^2},$$

where $\mathbf{A} := \mathbf{X}\mathbf{X}^\top + \frac{1}{2C}\mathbf{I}$.

Order of computation:

- first compute \mathbf{A}^{-1} , then \mathbf{w}_{RR} , and last $\text{RMSE}_{\text{loocv}}$

Proof

Recalling $\mathbf{w}_{RR} = A^{-1}Xy$ and denoting $\beta_i := \mathbf{x}_i^\top A^{-1} \mathbf{x}_i$, it is:

RMSE_{loocv}

$$\begin{aligned} &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^\top \left(A^{-1} + \frac{A^{-1} \mathbf{x}_i \mathbf{x}_i^\top A^{-1}}{1 - \mathbf{x}_i^\top A^{-1} \mathbf{x}_i} \right) (Xy - \mathbf{x}_i y_i) - y_i \right)^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^\top \mathbf{w}_{RR} + \frac{\beta_i \mathbf{x}_i^\top \mathbf{w}_{RR}}{1 - \beta_i} - \beta_i y_i - \frac{\beta_i^2}{1 - \beta_i} y_i - y_i \right)^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\left(1 + \frac{\beta_i}{1 - \beta_i} \right) \mathbf{x}_i^\top \mathbf{w}_{RR} - \left(\beta_i + \frac{\beta_i^2}{1 - \beta_i} + 1 \right) y_i \right)^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\mathbf{x}_i^\top \mathbf{w}_{RR} - y_i}{1 - \beta_i} \right)^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\mathbf{x}_i^\top \mathbf{w}_{RR} - y_i}{1 - \mathbf{x}_i^\top A^{-1} \mathbf{x}_i} \right)^2} \end{aligned}$$