

Chapter 5 Resampling

Overall Purpose of the Chapter

Resampling methods repeatedly draw (sub)samples from the **training data** and refit models to gain additional insight that is not available from a single model fit on the original data.

Main goals:

- Estimate **test error** (model assessment)
- Perform **model selection** (choose best level of flexibility / complexity)
- Estimate **uncertainty** / variability of parameter estimates or predictions

Two main families are covered:

1. **Cross-validation** → mainly used for **model assessment** and **model selection** (estimating test error)
2. **Bootstrap** → mainly used for estimating **standard errors** / uncertainty of estimates

5.1 Cross-Validation

Main Idea

Estimate how well a model will perform on **new/unseen data** (test error) using only the available training data.

5.1.1 Validation Set Approach (Hold-out / Train–Test split)

- Randomly split data into **training set** + **validation set**
- Fit model(s) on training set → evaluate on validation set (usually using MSE or misclassification rate)
- Problems:
 - High variability (depends heavily on which points go into validation)
 - Tends to **overestimate** test error (validation set is small → training set lacks data)
 - Only uses part of the data for training

5.1.2 Leave-One-Out Cross-Validation (LOOCV)

Special case of k-fold CV where **k = n**

- **LOOCV**: A special case of k-fold CV where $k=n$ (n = number of observations). For each of the n iterations, you leave out one data point, train on the remaining $n-1$, predict the left-out point, and compute its error. Average these errors for the CV estimate.

Procedure:

- For each $i = 1$ to n :

- Train on all data **except** observation i
- Predict \hat{y}_i using the left-out point
- Compute error on that single point: $MSE_i = (y_i - \hat{y}_i)^2$ or $Err_i = I(y_i \neq \hat{y}_i)$
- Final CV error:

LOOCV (regression):

$$CV_{(n)} = (1/n) \sum (y_i - \hat{y}_i)^2$$

LOOCV (classification):

$$CV_{(n)} = (1/n) \sum I(y_i \neq \hat{y}_i)$$

Shortcut formula (linear models / least squares / polynomials):

$$CV_{(n)} = (1/n) \sum [(y_i - \hat{y}_i) / (1 - h_i)]^2$$

where h_i = leverage of observation i

Advantages:

- Almost unbiased estimate of test error
- Uses almost all data for training each time

Disadvantages:

- Very high variance (predictions are highly correlated, each model uses $n-1 \approx n$ points)
- Computationally expensive unless shortcut formula is used

5.1.3 k-Fold Cross-Validation (most commonly used)

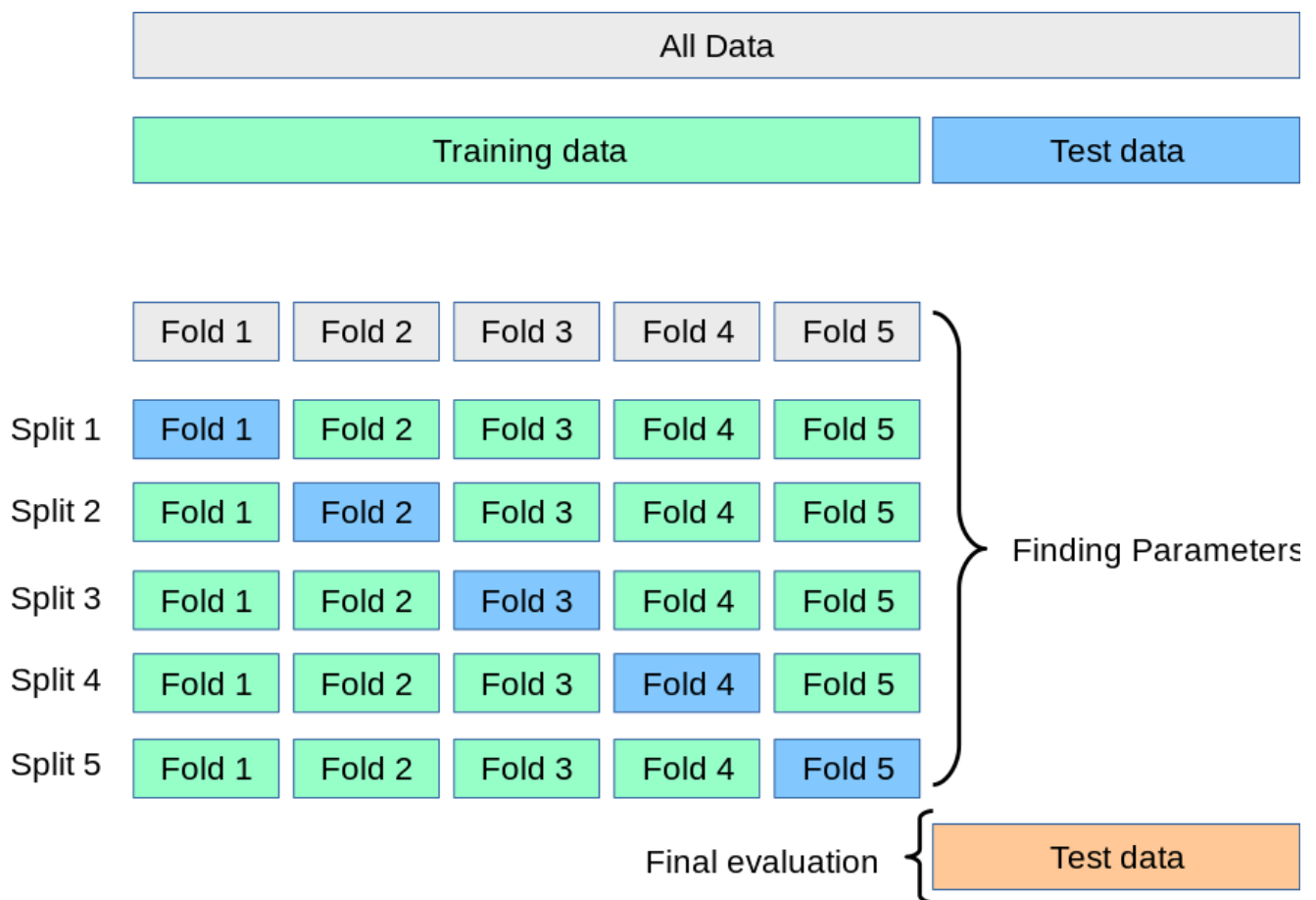
- Randomly divide data into k roughly equal-sized folds (usually $k = 5$ or $k = 10$)
- For each fold $j = 1$ to k :
 - Train on $k-1$ folds
 - Test on the held-out fold j
 - Compute MSE_j (or error rate on fold j)
- Final estimate:

k-fold CV:

$$CV_{(k)} = (1/k) \sum MSE_j$$

Advantages over LOOCV:

- Much lower variance
- Computationally much cheaper ($k \ll n$)
- $k = 5$ or 10 usually gives good **bias-variance trade-off**



Example

10-Fold CV: General k-fold with $k=10$. Split data into 10 equal folds; for each iteration, train on 9 folds (90% of data), test on the held-out fold (10%), compute error (e.g., MSE_j for fold j). Average over 10 folds.

Bias-variance trade-off summary

| METHOD | BIAS | VARIANCE | COMPUTATION | TYPICAL CHOICE |
|----------------|--------------|--------------|-------------|----------------|
| Validation set | high | high | low | — |
| LOOCV | very low | very high | high | rare |
| 5-fold CV | low–moderate | moderate | moderate | very common |
| 10-fold CV | very low | low–moderate | higher | very common |

- **Bias:** How much the method systematically over- or under-estimates the true test error or variability.
 - LOOCV: Very low bias because each model is trained on nearly all data ($n-1 \approx n$), so it's close to the full model's performance.
 - 10-Fold CV: Slightly higher bias than LOOCV (trains on 90% of data), but still low—especially for larger n . It can overestimate test error a bit more than LOOCV for

small datasets.

- **Bootstrap:** Low bias for estimating variability if the statistic is unbiased, but it can underestimate variance in small samples (since samples are with replacement, leading to 63% unique data per bootstrap on average).
- **Variance:** How much the estimate fluctuates across different data splits or resamples.
 - **LOOCV:** High variance because the n models are highly correlated (each overlaps by $n-2$ points), so the CV error can swing based on outliers.
 - **10-Fold CV:** Lower variance than LOOCV (fewer, less correlated models), making it more stable.
 - **Bootstrap:** Moderate to low variance if B is large; it's robust because resamples are independent draws.

5.1.4 Bias-Variance Trade-off in k-fold CV

- k increase \rightarrow bias decrease but variance increase
- | K | Bias | Variance |
| Small K | High | Low |
| Large K | Low | High |
- $k = 5$ or 10 usually preferred in practice (good compromise)

5.1.5 Cross-Validation for Classification

Same logic applies, just replace MSE with misclassification error rate (or 0-1 loss):

$\text{Err}_j = (\text{number of misclassifications in fold } j) / (\text{size of fold } j)$

CV error = average Err_j over k folds

5.2 The Bootstrap

"to pull yourself up by your bootstraps"

Goal: Estimate **standard error** (or confidence intervals) of any statistic / estimator **using only the original data.**

Used for:

- Variance
- Mean
- Model performance

Sampling with resampling = Randomly selecting data and allowing for duplicates.

Bootstrapping consist of 3 steps:

- make a bootstrapped dataset
- calculate mean, median, ...
- keep track of that calculation

Core idea: Treat the **original sample** as if it were the population → repeatedly draw samples **with replacement** from it.

Procedure (basic bootstrap):

1. Original dataset Z with n observations
2. Draw B bootstrap samples Z^1, Z^2, \dots, Z^{*B} (each of size n , sampling **with replacement**)
3. Compute the statistic/estimate θ^{*b} on each bootstrap sample $b = 1 \dots B$
4. Bootstrap estimate of standard error:

$$SE_B(\hat{\theta}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^B \left(\hat{\theta}^{*b} - \bar{\theta}^* \right)^2}$$

where $\bar{\theta}^* = \frac{1}{B} \sum_{b=1}^B \hat{\theta}^{*b}$. Typically $B=1000+$ for stability.

Ex: $SE = 0.018 \rightarrow$ this 0.082 estimate has an uncertainty of about ± 0.018 (very roughly speaking).

Most common uses:

- Standard error of regression coefficients
- Standard error of a complicated estimator (e.g. best α in portfolio allocation)
- Accuracy of any fitted model / prediction method

| Aspect | LOOCV | 10-Fold CV | Bootstrap   |
|---------------------|--|---|---|
| Primary Use | Test error estimation; model selection in small data | General test error; hyperparameter tuning (e.g., GridSearchCV) | Uncertainty (SEs, CIs); works for any statistic, even non-parametric |
| Best For Datasets | Small n (low bias helps); linear models (shortcut) | Medium-large n; any model | Any n; when variance/SE is key (e.g., finance, biostats) |
| Pros | Unbiased; uses max data per fit | Good bias-variance balance; stable; fast | Flexible (any estimator); quantifies uncertainty; bias correction possible (e.g., BCa bootstrap) |
| Cons | High variance; slow for non-linear models | Slight bias; depends on fold randomness (repeat for stability) | No direct test error; underestimates variance in dependent data; high compute for B large |
| Real-World Examples | Medical studies with few patients (e.g., predict disease from 50 scans); polynomial regression | Kaggle competitions (tune models on 10k+ rows); deep learning (though often 5-fold for speed) | Finance (SE of portfolio α , as in Figure 5.10); hypothesis testing (bootstrap p-values); ML feature importance |
| ISLR Figure Ties | Fig 5.4/5.6: More variable than k-fold | Fig 5.6/5.8: Tracks test error U-shape well in classification | Fig 5.9-5.11: Mimics true sampling dist. for α in investments |
| Software Tips | scikit-learn: <code>LeaveOneOut()</code> ; use with <code>LinearRegression</code> for speed | scikit-learn: <code>KFold(n_splits=10)</code> ; default for <code>GridSearchCV</code> | scikit-learn: <code>resample()</code> or boot library; R's <code>boot</code> package  |