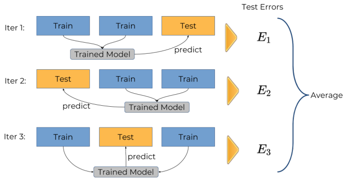
Introduction to Machine Learning

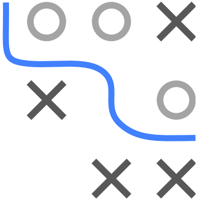
Evaluation
Resampling 1



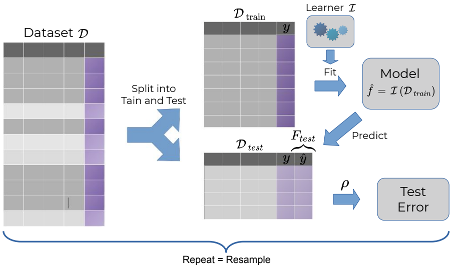
Learning goals

- Understand how resampling techniques extend the idea of simple train-test splits
- Understand the ideas of cross-validation, bootstrap and subsampling

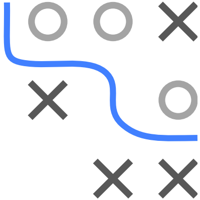
RESAMPLING



- **Goal**: estimate $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n, \rho_L) = \mathbb{E}\left[L(y, \mathcal{I}(\mathcal{D}_{\text{train}}, \boldsymbol{\lambda})(\mathbf{x}))\right]$.
- Holdout: Small trainset = high pessimistic bias; small testset = high var.
- Resampling: Repeatedly split in train and test, then average results.
- Allows to have large trainsets large (low pessimistic bias) since we use $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n_{\mathtt{train}}, \rho)$ as a proxy for $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n, \rho)$)
- And reduce var from small testsets via averaging over repetitions.



RESAMPLING STRATEGIES



- Represent train and test sets by index vectors::
  $J_{\text{train}} \in \{1, \dots, n\}^{n_{\text{train}}}$ and $J_{\text{test}} \in \{1, \dots, n\}^{n_{\text{test}}}$
- Resampling strategy = collection of splits:

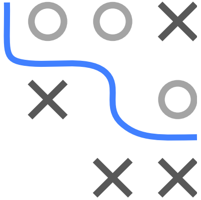
$$\mathcal{J} = \left((J_{\text{train},1}, J_{\text{test},1}), \dots, (J_{\text{train},B}, J_{\text{test},B}) \right).$$

- Resampling estimator:

$$\begin{aligned}\widehat{\mathrm{GE}}(\mathcal{I}, \mathcal{J}, \rho, \boldsymbol{\lambda}) = \operatorname{agr}\Big(& \rho\Big(\mathbf{y}_{J_{\text{test},1}}, \boldsymbol{F}_{J_{\text{test},1}, \mathcal{I}(\mathcal{D}_{\text{train},1}, \boldsymbol{\lambda})}\Big), \\ & \vdots \\ & \rho\Big(\mathbf{y}_{J_{\text{test},B}}, \boldsymbol{F}_{J_{\text{test},B}, \mathcal{I}(\mathcal{D}_{\text{train},B}, \boldsymbol{\lambda})}\Big)\Big),\end{aligned}$$

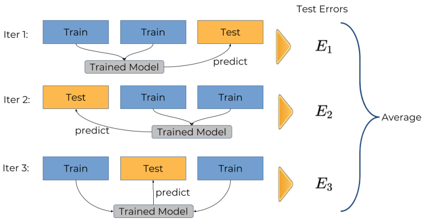
- Aggregation agr is typically "mean" and $n_{\text{train}} \approx n_{\text{train},1} \approx \dots \approx n_{\text{train},B}$.

CROSS-VALIDATION

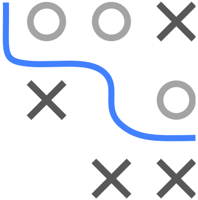


- Split the data into k roughly equally-sized partitions.
- Each part is test set once, join $k-1$ parts for training.
- Obtain k test errors and average.
- Fraction $(k-1)/k$ is used for training, so 90% for 10CV
- Each observation is tested exactly once.

Example: 3-fold CV

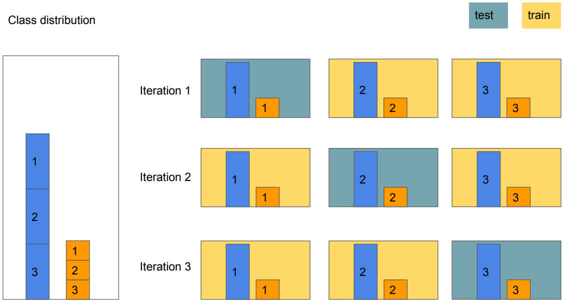


CROSS-VALIDATION - STRATIFICATION



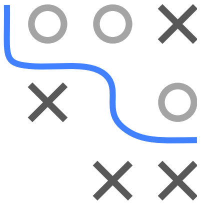
- Used when target classes are very imbalanced
- Then small classes can randomly get very small in samples
- Preserve distrib of target (or any feature) in each fold
- For classes: simply CV-split the class data, then join

Example: stratified 3-fold cross-validation



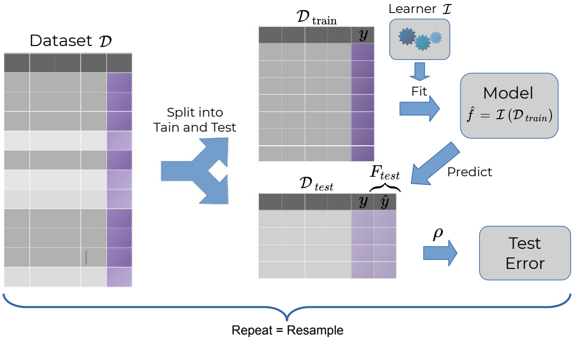
CROSS-VALIDATION

- 5 or 10 folds are common.
- $k = n$ is known as "leave-one-out" CV (LOO-CV)
- Bias of $\widehat{\mathrm{GE}}$: The more folds, the smaller. LOO nearly unbiased.
- LOO has high var, better many folds for small data but not LOO
- Repeated CV (avg over high-fold CVs) good for for small data.



SUBSAMPLING

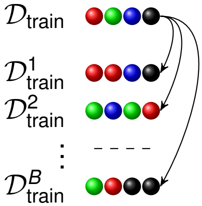
- Repeated hold-out with averaging, a.k.a. Monte Carlo CV.
- Typical choices for splitting: $\frac{4}{5}$ or $\frac{9}{10}$ for training.



- Smaller subsampling rate = larger pessimistic bias
- More reps = smaller var

BOOTSTRAP

- Draw B trainsets of size n with replacement from orig $\mathcal{D}$
- Testsets = Out-Of-Bag points: $\mathcal{D}^b_{\text{test}} = \mathcal{D} \setminus \mathcal{D}^b_{\text{train}}$



- Similar analysis as for subsampling
- Trainsets contain about 2/3 unique points:
  $1 - \mathbb{P}((\mathbf{x}, y) \notin \mathcal{D}_{\text{train}}) = 1 - \left(1 - \frac{1}{n}\right)^n \overset{n\to\infty}{\longrightarrow} 1 - \frac{1}{e} \approx 63.2\%$
- Replicated train points can lead to problems and artifacts
- Extensions B632 and B632+ also use trainerr for better estimate when data very small

LEAVE-ONE-OBJECT-OUT

- Used when we have multiple obs from same objects, e.g., persons or hospitals or base images
- Data not i.i.d. any more
- Data from same object should **either** be in train **or** testset
- Otherwise we likely bias $\widehat{\mathrm{GE}}$
- CV on objects, or leave-one-object-out