How do we obtain reliable estimates of performance measures?

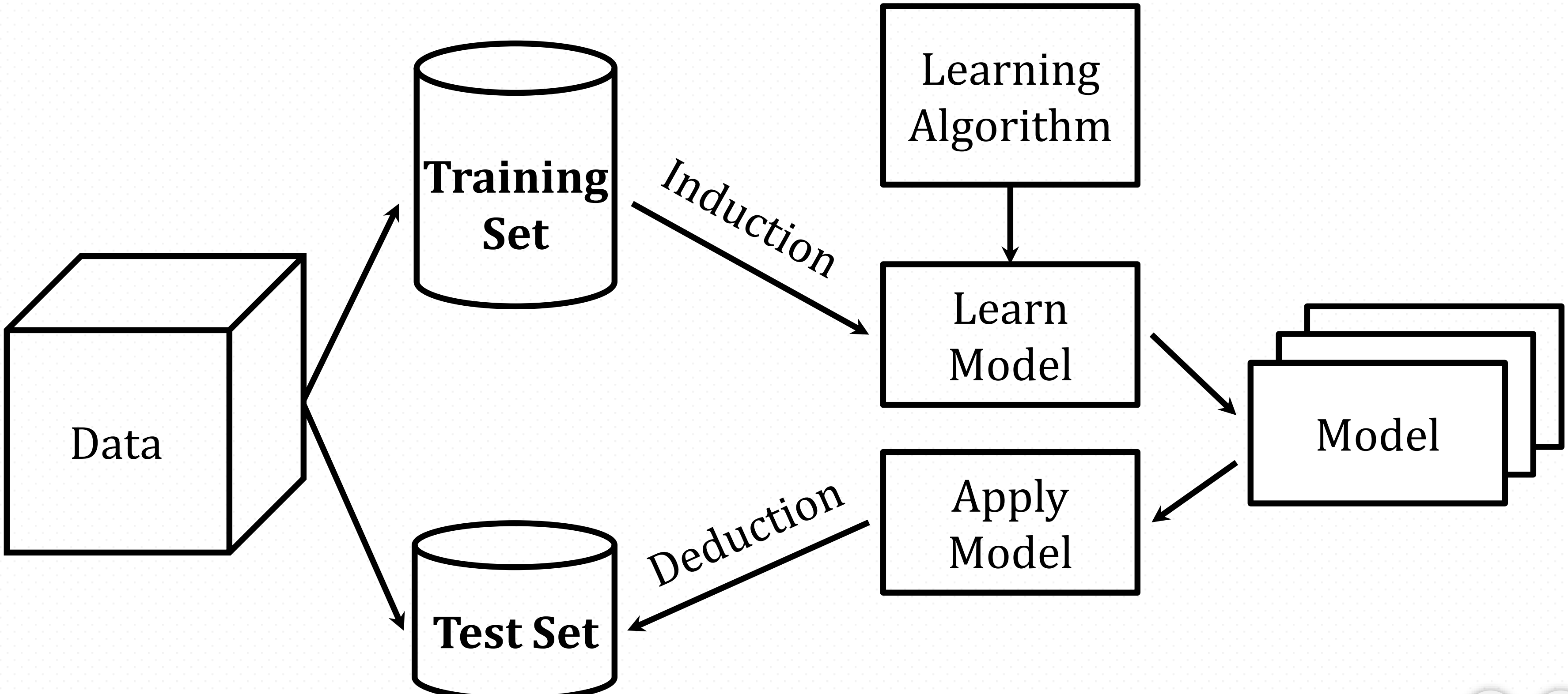
Estimating Model Performance

- How do we estimate performance measures?

- Error on training data?
  - Also called resubstitution error.
  - Not a good indicator of the performance on future data.

- Simple solution
  - Spit the available data into training and testing sets.

Training and Testing Sets



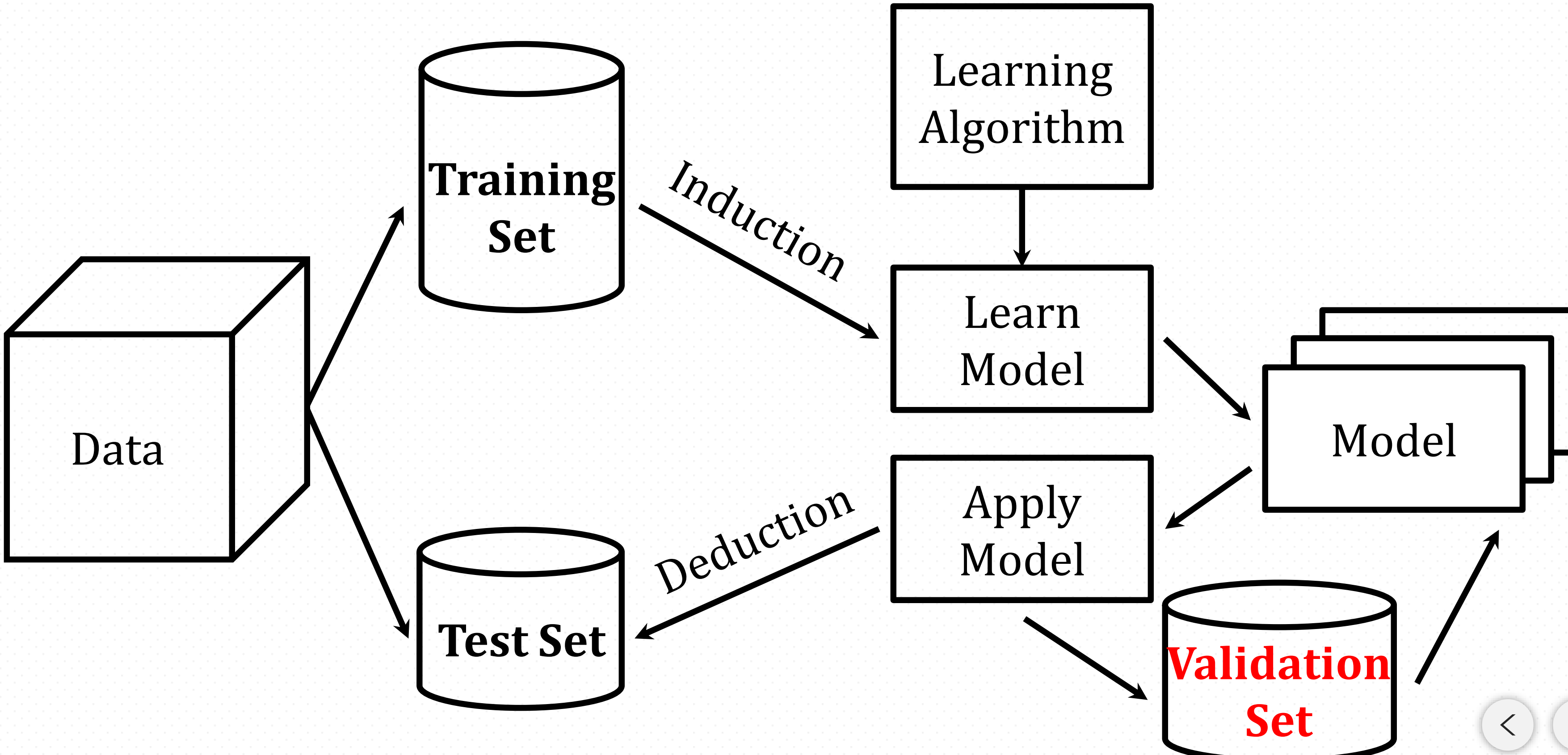
Avoiding Data Snooping

- It is important that the test data is not used in any way to create the classifier.
- Some learning schemes operating in two stages
  - Stage 1: builds the basic structure
  - Stage 2: optimizes parameter settings
- The test data cannot be used for parameter tuning.
- Proper procedure uses three sets: training data, validation data, and test data.

Validation Data

- A validation dataset is a subset of the data used to tune parameters.
- Typically used when an appropriate model needs to be chosen from several rivaling approaches.

Validation Data



Methods of Estimating Performance

- Holdout
  - Reserve ½ for training and ½ for testing.
  - Reserve 2/3 for training and 1/3 for testing.
- Random subsampling
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on $k-1$ partitions, test on the remaining one
  - Leave-one-out: $k=n$

Methods of Estimating Performance

- Holdout
  - Single holdout
  - Repeated holdout
- Cross validation
  - k-fold validation
  - Leave-one-out validation
- Stratified sampling
- Bootstrap

Holdout Estimation

Key Idea:

Reserve a certain amount of data for testing and use the remainder for training.

Problems:

- For small or "unbalanced" datasets, instances might not be representative.
- The data used for training and testing may vary significantly.

Stratified Holdout

Generate holdout using *stratified sampling*.

- Generates new subsets of instances with an approximately equal proportions of classes.

Ensures that the classes are equally represented in the samples.

Repeated Holdout

- Repeated holdout, or “random subsampling,” improves the reliability of the holdout estimate by repeating the process with different subsamples.
  - In each iteration, a certain proportion of data is randomly selected for training.
  - The error rates on different iterations are averaged to yield an overall error rate.
- Problem: overlapping test sets.

Cross-Validation

- Cross-validation ensures non-overlapping test sets.
- In k-fold cross-validation:
  - The data is split into k stratified subsets of equal size.
  - Each of the k subsets is used for testing and the combination of the rest for training.
- The error estimates are averaged across each of the k folds.

Example of Cross-Validation

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10

1. Divide a dataset into k folds.

Example of Cross-Validation

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10

1. Divide a dataset into k folds.
2. Use one subset for testing and the remainder for training.

Example of Cross-Validation

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10

1. Divide a dataset into *k* folds.
2. Use one subset for testing and the remainder for training.
3. Iterate.

Example of Cross-Validation

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10

1. Divide a dataset into k folds.
2. Use one subset for testing and the remainder for training.
3. Iterate.
4. Average the error rates over all k folds.

Properties of Cross-Validation

- Cross-validation uses sampling without replacement.
  - The same instance, once selected, cannot be selected again for a particular training/testing set.
- Computationally expensive.
- Variance tends to be high.

The Bootstrap

- The bootstrap uses sampling with replacement to form the training set.
  - Sample a dataset of n instances n times with replacement to form a new dataset of n instances.
  - Use this data as the training set.
  - Use the instances from the original dataset that don't occur in the new training set for testing.

The Bootstrap

- An instance has a probability of $1 - 1/n$ of not being picked for training.
- Thus, its probability of ending up in the test data is:

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

- This means the training data will contain approximately 63.2% of the instances.

Estimating Error using the Bootstrap

- The error estimate on the test data will be very pessimistic, since training was on just ~63% of the instances.
- Therefore, combine it with the resubstitution error:
$$err = 0.632 \times e_{\text{test instances}} + 0.368 \times e_{\text{training instances}}$$
- The resubstitution error gets less weight than the error on the test data.

Properties of the Bootstrap

- For small sample size n, bootstrap will have much smaller variability than the cross-validation estimate.
- Bootstrap and CV estimates will generally be close for large sample sizes.
  - Their ratio will approach unity as the sample size approaches infinity.