Pattern Recognition

Lecture 20. Cross Validation

Dr. Shanshan ZHAO

shanshan.zhao@xjtlu.edu.cn

School of AI and Advanced Computing

Xi'an Jiaotong-Liverpool University

Academic Year 2021-2022

Introduction

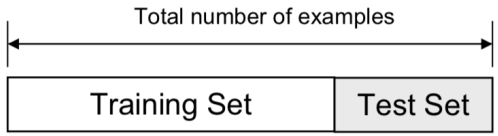
- Two issues in pattern recognition techniques
  - Model Selection
    - How do we select the “optimal” parameter(s) for a given classification problem?
  - Validation
    - Once we have chosen a model, how do we estimate its true error rate?
- If we had access to an unlimited number of examples, these questions would have a straightforward answer.
  - Choose the model that provides the lowest error rate on the entire population.
- However, in real applications only a finite set of examples is available
  - This number is usually smaller than we would hope for!
  - Why? Data collection is a very expensive process

The techniques to make the best use of your (limited) data for

- Training
- Model selection
- Performance estimation

1. The holdout method

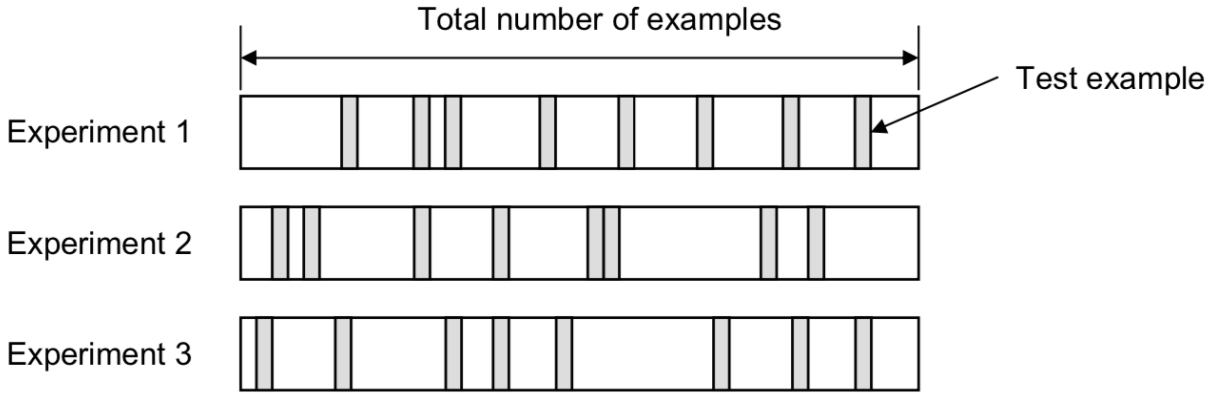
- Split dataset into two groups
  - **Training set:** used to train the classifier
  - **Test set:** used to estimate the error rate of the trained classifer



- The holdout method has two basic drawbacks
  - In problems where we have a sparse dataset we may not be able to afford the “luxury” of setting aside a portion of the dataset for testing
  - Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “unfortunate” split
- The limitations of the holdout can be overcome with a family of re- sampling methods at the expense of higher computational cost
  - Cross Validation
    Random Subsampling; K-Fold Cross-Validation; Leave-one-out Cross-Validation
  - Bootstrap

2. Random Subsampling

- Random Subsampling performs K data splits of the entire dataset.
  - Each data split randomly selects a (fixed) number of examples without replacement
  - For each data split we retrain the classifier from scratch with the training examples and then estimate e_i with the test examples



- The true error estimate is obtained as the average of the separate estimates e_i

$$E = \frac{1}{K} \sum_{i=1}^{K} e_i$$

3. K-Fold Cross-validation

- Create a K-fold partition of the the dataset
  - For each of K experiments, use K-1 folds for training and a different fold for testing

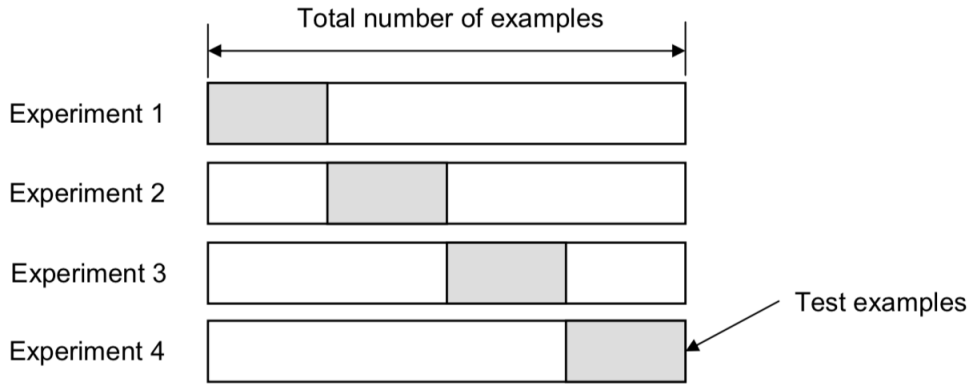


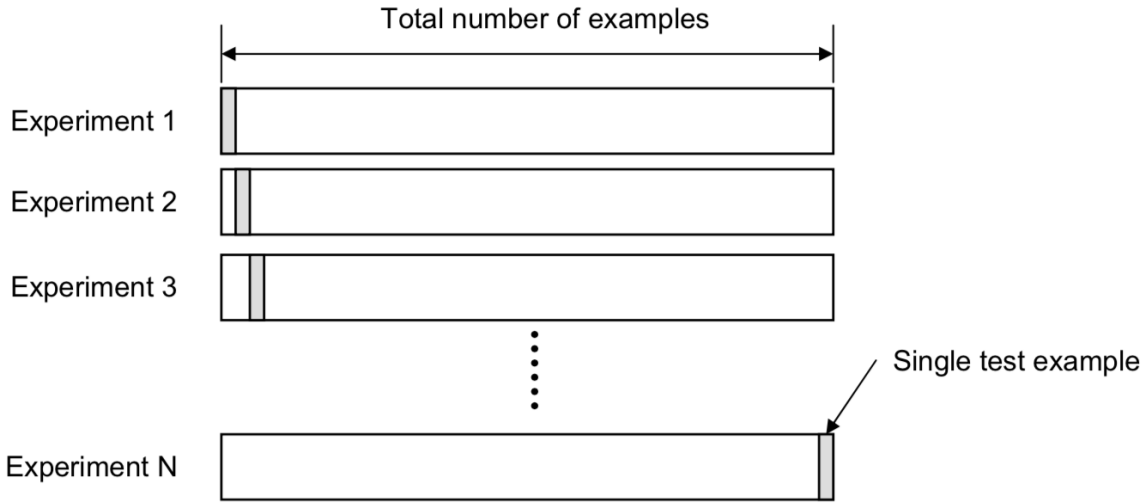
Figure: When $K = 4$

- K-Fold Cross validation is similar to Random Subsampling
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- As before, the true error is estimated as the average error rate on test examples

$$E = \frac{1}{K} \sum_{i=1}^{K} e_i$$

4. Leave-one-out Cross Validation

- Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples
  - For a dataset with N examples, perform N experiments
  - For each experiment use N-1 examples for training and the remaining example for testing



- As usual, the true error is estimated as the average error rate on test examples

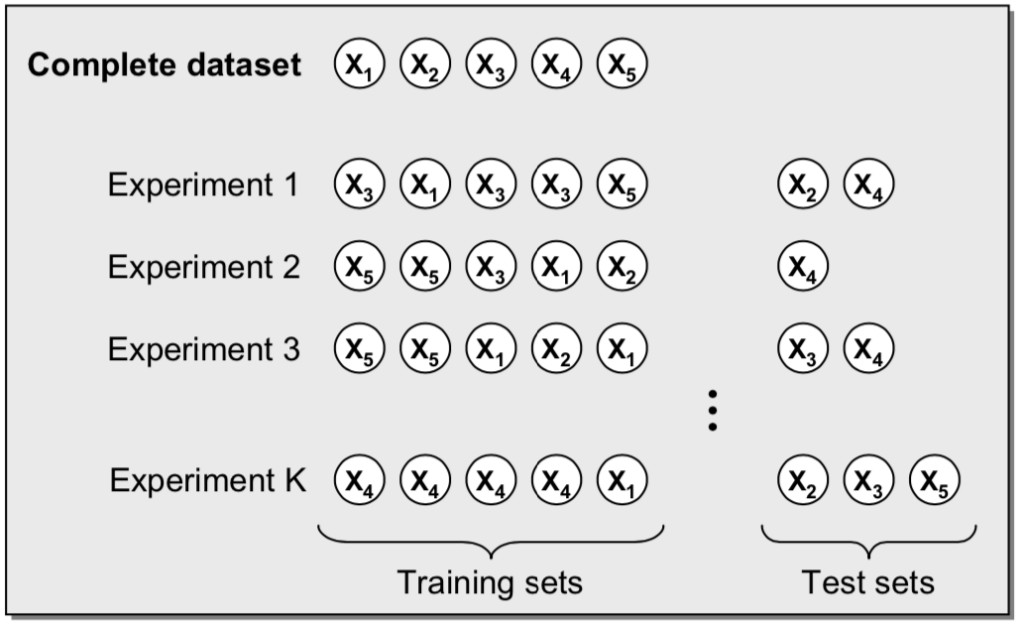
$$E = \frac{1}{K} \sum_{i=1}^{K} e_i$$

How to choose K?

- With a large number of folds
  - The bias of the true error rate estimator will be small (the estimator will be very accurate)
  - The variance of the true error rate estimator will be large
  - The computational time will be very large as well (many experiments)
- With a small number of folds
  - The number of experiments and, therefore, computation time are reduced
  - The variance of the estimator will be small
  - The bias of the estimator will be large (conservative or smaller than the true error rate)
- In practice, the choice of the number of folds depends on the size of the dataset
  - For large datasets, even 3-Fold Cross Validation will be quite accurate
  - For very sparse datasets, we may have to use leave-one-out in order to train on as many examples as possible
- A common choice for K-Fold Cross Validation is K=10

5. The bootstrap

- The bootstrap is a resampling technique with replacement
  - From a dataset with N examples
    - Randomly select (with replacement) N examples and use this set for training
    - The remaining examples that were not selected for training are used for testing
      This value is likely to change from fold to fold
  - Repeat this process for a specified number of folds (K)
  - As before, the true error is estimated as the average error rate on test examples



Three-way data splits

- If model selection and true error estimates are to be computed simultaneously, the data needs to be divided into three disjoint sets [Ripley, 1996]
  - Training set: a set of examples used for learning: to fit the parameters of the classifier
  - Validation set: a set of examples used to tune the parameters of a classifier
  - Test set: a set of examples used only to assess the performance of a fully-trained classifier
- Why separate test and validation sets?
  - The error rate estimate of the final model on validation data will be biased (smaller than the true error rate) since the validation set is used to select the final model
  - After assessing the final model on the test set, YOU MUST NOT tune the model any further!

Three-way data splits

- Procedure outline

  1. Divide the available data into training, validation and test set
  2. Select architecture and training parameters
  3. Train the model using the training set
  4. Evaluate the model using the validation set
  5. Repeat steps 2 through 4 using different architectures and training parameters
  6. Select the best model and train it using data from the training and validation sets
  7. Assess this final model using the test set

Reference I

Thank You !

$\mathcal{Q}$ & $\mathcal{A}$