Generalization, Model Validation and Optimization
Ahmed Eleish
Data Analytics ITWS-4600/ITWS-6600/MATP-4450/CSCI-4960
February 14th 2025

Tetherless World Constellation
Rensselaer Polytechnic Institute

Errors in Classification

- We've seen classification errors working with the iris/abalone classification examples
- Now we will take a look at what are the possible errors that take place in classification.
- **If the predicted class label is different from the actual class label (true class) then there is an error with that classification**.
- In classification, the model's output is the predicted class label for the input variables and the true class label is the target.

Misclassification Error

- The error rate is the percentage of errors made over the entire dataset

- Error rate is also known as the misclassification rate or simply called the error.

- Error = (*Number of Misclassifications*)/(Total Number of Samples)

- Accuracy = (*Number of Correct Classifications*)/(Total Number of Samples)

Training, test and validation sets

- Training: subset of dataset used as input to the model's training algorithm

- Validation: subset used to evaluate models during training

- Test: subset used to test the final model

e.g. the training set (70%) is used to train multiple models (different features, parameters, etc.) and the validation set (20%) is used to compare and select the best performing model. The test set (10%) is then used to evaluate the selected model.

Terminology Confusion!

- 'Test' and 'validation' are used interchangeably in academia and industry!

- That's fine… make sure to know the proper use.

- In this class:

  - While training, the dataset is split into 2 subsets: training & test.

  - A validation set may be kept out of the training process for final evaluation, but this is not required for your labs/assignments.

https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets

Errors

- The error on the training data is called as the "**training error**"
- The error on the test data is referred to as the **"test error"**
- **The error on the test data is a good indication of how well the classifier will perform on new data and this is known as the generalization**.
- If the classifier performs well on the new data, then it is a good generalization. Generalization refers to how well the model is performing on the new data (**data not used to train the model**)

Test error : Generalization error

- If the model generalizes well, then it will perform well on the new data sets that has the *similar structure* to the training data..

- Since the Test error is an indication of how well the model generalizes to new data, *the test error also called the generalization error.*

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition

Overfitting

- Another related concept to Generalization is "overfitting".

- If the model has very low training error but it has high generalization error, then it is over fitting.
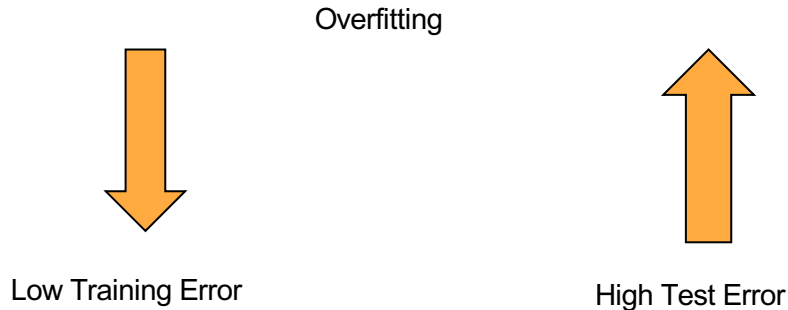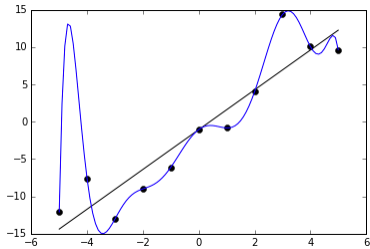
Overfitting

Low Training Error

High Test Error

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition
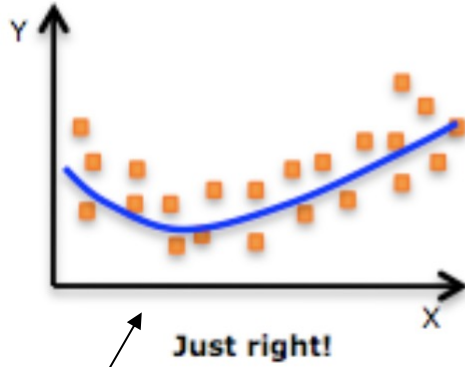
Overfitting

- This is a good indication that the model may have learned to **_model the noise_** in the training data, instead of the learning from the underlying structure of the data.
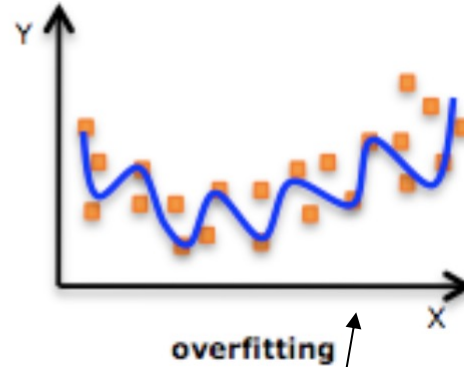- Overfitting is an indication of poor generalization.

Image/Photo Credit:
https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitted_Data.png

Image/Photo Credit:
http://pingax.com/regularization-implementation-r/

Just right!
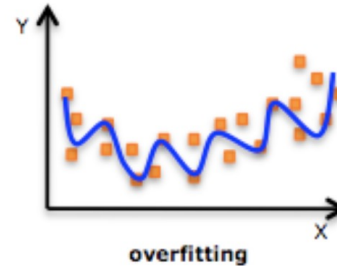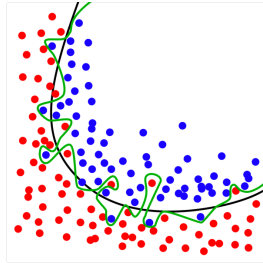
overfitting

Model is fitting to
the structure of the data

Model is fitting to
the noise of the data

Image/Photo Credit: http://pingax.com/regularization-implementation-r/

Underfitting

- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data.
- In other words, **underfitting take place when the model has not properly learned the structure of the data**.

Underfitting

Image/Photo Credit: http://pingax.com/regularization-implementation-r/
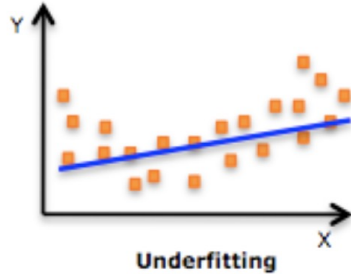
Robustly Validating Models

- There are several ways to create the evaluate/validate models

  - K-fold Cross validation

  - Monte Carlo Cross validation

  - Leave-One-Out Cross validation

K-fold Cross Validation

- In k-fold cross validation, the data are segmented in to k number of **disjoint partitions**.

- During each iteration, one partition is used as the test set and the remaining k-1 (combined) for training; The process is repeated k times.

- Each time using a different partition for testing, so that each partition is used exactly one time for the validation.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Monte Carlo Cross Validation (Repeated random sub-sampling)

- In Monte Carlo cross validation, the dataset is split into training/test sets over n iterations with the samples in each selected at random.

- The size of each partitions may be constant or vary over the iterations.

- Commonly used in research, considered robust because of the averaging effect over multiple iterations.

- Downside: since selection is random, some observations may not end up in test sets and some may be oversampled

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Rensselaer

Tetherless World Constellation

13

Leave One Out Cross Validation (LOOCV)

- For as many iterations as there are observations, drop one observation and used all the others for training; test one the 1 observation and average at the end.

- Depending on the size of the dataset, may be computationally expensive.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

https://rpi.box.com/s/8deje65k0xrhax597o8ahj1tv6bl396f

Optimizing kNN Models

- The parameter k represents the number of nearest neighbors used by the algorithm

- Rule of thumb: $k = n^{1/2} = \sqrt[2]{n}$

- Finding the optimal value for k

    - For a range of k values, train a kNN model and calculate classification accuracy

    - Select k value from best performing model

Optimizing kNN Models

- Finding the optimal value for *k*

    ○ For a range of *k* values, train a
    kNN model and calculate
    classification accuracy

   ○ Select *k* value from best
    performing model
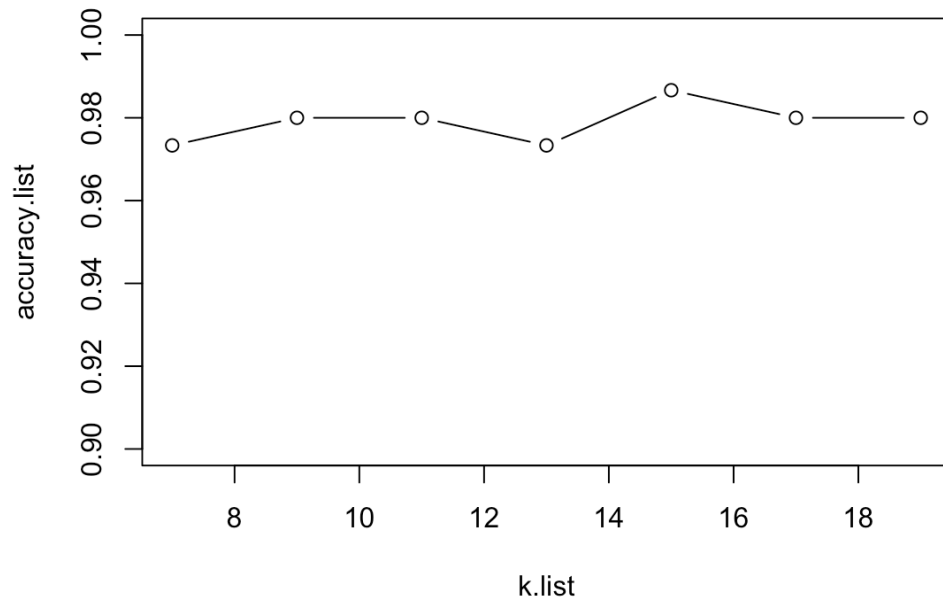
Optimizing K-Means Models

- The parameter K represents the number of clusters to be identified by the algorithm

- Depends on background knowledge/research question

- Finding the optimal value for K

  - For a range of K values, train a K-Means model and calculate within cluster sum of squares (WCSS)

  - Select K value where after which the decrease in WCSS diminishes

  - This is called the elbow method

Optimizing K-Means Models

- Finding the optimal value for *K*

  - For a range of *K* values, train a K-Means model and calculate within cluster sum of squares (WCSS)

  - Select *K* value where after which the decrease in WCSS diminishes

  - This is called the elbow method
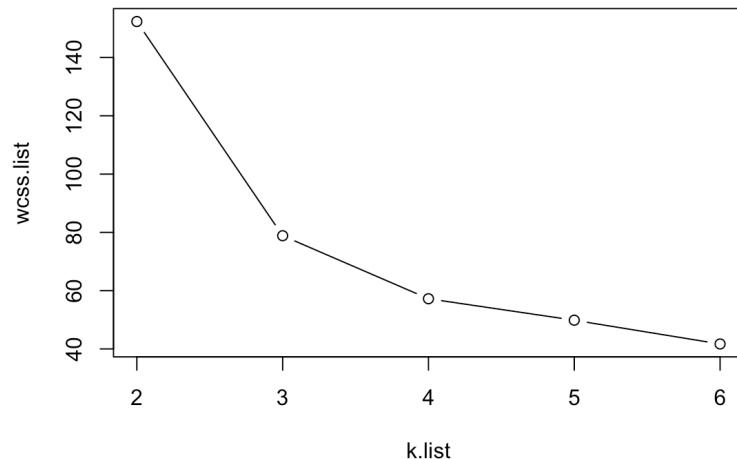
Thanks!
Have a great weekend!

Work on your assignment/project proposal!!!