# Hands on machine learning

Ziad Yasser Abdullah
Omar Samir Abosamaha

ULM university
Institute of Neural Information Processing
Supervised by Prof. Dr. Friedhelm Schwenker

March 30, 2020

# Outline

- Chapter 1
  - supervised vs unsupervised
  - Online vs Batch
  - Instance based vs Model based
  - Challenges
- Chapter 2
  - Project description
  - Work-Flow
  - Learning points and Handy tips
- Chapter 3
  - Types of Classification
  - Performance measures
- Chapter 4
  - Training Models
- Chapter 5
  - Support Vector Machines
- Chapter 6
  - Decision Trees

# Chapter 1

topics discussed in chapter 1:

- Supervised Learning vs Unsupervised Learning
- online learning vs batch learning.
- Instance based Learning vs Model Based Learning.
- challenges.

# Chapter 1
## supervised vs unsupervised

Supervised : Feeding the algorithm instances with Labels.

- Regression.
- classification.

Unsupervised : Feeding the algorithm instances without Labels.

- Clustering.
- Visualization.

SemiSupervised : Usually lot of unlabeled data little bit of labeled data.

Reinforocement : The agent chooses a policy and observe his performance based on rewards and punishments.

- AlphaGO robot.

# Chapter 1
# Online vs Batch

Batch Learning : Offline learning, the model cannot learn incrementally using the whole set every time.

Online Learning : Incremental learning where data becomes available in a sequential order and is used to update our best predictor for future data at each step.

# Chapter 1
# Instance based vs Model based

Instance Based : Know the examples by heart then generalize using similarity measure.

Model Based : Builds a model of the data given then use that model for predictions.

# Chapter 1
# What can go wrong?

Bad data:

- Insufficient Quantity of training data.
- The unreasonable effectiveness of training data.
- Non-representative training data.
- Poor quality of training data.
- Irrelevant features.

Bad Algorithm:

- Over-fitting the training data.
  - can be fixed by regularization.
- Under-fitting the training data.
  - can be fixed by more complex model or more training data.

# Chapter 2

## Chapter 2 was mainly a coding project

### Chapter 2 in points:

- What was the problem ?
- Work-Flow.
- Learning points and handy tips.

# Chapter 2
# Project description

## What was the problem?

- Building a model of housing prices in California using the California census data.
- The model has to predict the price of a house in a specific district.
- The output of our model is to be fed to another machine learning algorithm to decide whether or not it is worth investing in.

# Chapter 2
# Work-Flow

1. Load the data.
2. Understand the data set we have.
3. Prepare the data for the ML algorithm.
4. Select a model and train it.
5. Fine tune your model.
6. Present your solution.
7. Launch, monitor, and maintain your system.

# Chapter 2
# Learning points and Handy tips

1. Try to automate as much as possible for example: load the data by code.
2. Splitting the data into.
   - Test set (Stratified).
   - Training set.
   - Validation set.
3. Use feature scaling.
4. Save the models that you experiment.
5. Kaggle seems a lovely website to practice.

# Chapter 3
# Classification

Topics discussed:

- Types of Classification.
- Performance measures.

# Chapter 3
# Classification

## Types of Classification:

- Binary Classification.
  - To classify the elements of a given set into two groups.
    - Stochastic Gradient Descent (SGD).
    - Support Vector Machines (SVM).
- Multi-class Classification.
  - Handling multiple classes either:
    1. Directly.
    2. Indirectly.
       - I One-Versus-All.
       - II One-Versus-One.
- Multi-label Classification.
  - The classification will output multiple classes.
- Multi-output Classification.
  - Generalization of Multiple Classification where each label can be multi-class.

# Chapter 3
# Performance measures

## Performance measures

**Confusion Matrix** : is a table with two rows and two columns that reports the number of false positives (FP), false negatives(FN), true positives(TP), and true negatives(TN). This allows more detailed analysis than mere proportion of correct classifications (accuracy).

- Precision: The accuracy of the positive prediction.
  $= TP/(TP + FP)$
- Recall: The ratio of positive instances that are correctly detected by the classification.
  $= TP/(TP + FN)$

**ROC Curve** : Plots the TPR vs the FPR.

# Chapter 4
# Training Models

Topics discussed:

- Linear Regression.
- Polynomial Regression.
- Generalization errors.
- Ridge Regression.
- Lasso Regression.
- Elastic Net Regression.
- Logistic Regression.
- Soft Max Regression.

# Chapter 4
# Training Models

## Linear Regression

can be trained in two ways either direct equation or iterative optimization.

## Gradient Descent :

-Learning step should not be too small nor too big.

-Features should have the same scale.

- Batch Gradient Descent :
  -Takes the whole patch of the data at each GD step.
- Stochastic Gradient Descent :
  -Takes a random instance in the training set at each GD step.
- Mini-Batch Gradient Descent :
  -Takes small random sets of instances in the training set at each GD step.

# Chapter 4
# Training Models

## Summary for Linear Regression

| Algorithm | Large m | Out-of-core | Large n | Scaling required |
|---|---|---|---|---|
| Normal Equation | Fast | No | Slow | No |
| SVD | Fast | No | Slow | No |
| Batch GD | Slow | No | Fast | Yes |
| Stochastic GD | Fast | Yes | Fast | Yes |
| Mini-batch GD | Fast | Yes | Fast | Yes |

Table: Comparison of algorithms for Linear Regression

## Polynomial Regression

1. Simply add powers of each feature as a new feature.
2. Run linear regression.

# Chapter 4
# Generalization Error

Generalization errors usually come from 3 types:

Bias:
Due to wrong assuming and leads to underfitting.

Variance:
Due to the model'sensitivity and leads to overfitting.

Irreducible error:
Due to the noisiness of the data itself.

## Solution ?

Regularization!!!

# Chapter 4
# Training Models

## Regularization for iterative Algorithm (GD)

**Early Stop:** Stopping the training as soon as the validation error reaches a minimum.

# Chapter 4
# Training Models

## Regularized linear Regressions

regularization is typically achieved by constraining the weights of the model.

Ridge Regression :

-Regularization term $((\|w\|_2)^2)/2$ is added to the cost function.
-The hyperparameter $\alpha$ controls how much you want to regularize the model.
-Can be computed by performing either.

- closed-form equation.
- Gradient Descent.

Lasso Regression :- Tends to completely eliminate the weights of the least important features.

Elastic Net :- Mix between Ridge Regression and Lasso Regression with mix ratio r.

# Chapter 4
# Training Models

## Logistic Regression

-Can be used for classification.
-Just like linear regression it computes a weighted sum of input features but instead of outputting the result directly it outputs the logistics of this results.

## Softmax Regression

-A generalization for Logistic Regression to support multiple classes directly without having to train and combine multiple binary classifiers.

# Chapter 5
# Support Vector Machines

Topics discussed:

- Linear Classification.
- Non-Linear Classification.
- SVM Regression.

# Chapter 5
# Support Vector Machines

## Linear Classification

- Hard margin Classification.
  - only when classes are linearly separable.
  - sensitive to outliers.
- Soft margin Classification.
  - Can be controlled by hyparameter C.
  - a smaller C value leads to a wider street but more margin violations.

# Chapter 5
# Support Vector Machines

## Non-Linear Classification

- Polynomial Kernel.
  - Same results of adding many polynomial features.
  - It is advised to GridSearch the best hyper-parameter.
- Gaussian RBF kernel.
  - just like adding many similarity features.
  - reducing the gamma leads to over-fitting the model.

# Chapter 5
# Support Vector Machines

| Class | Time complexity | Out-of-core | Scaling | Kernel |
|---|---|---|---|---|
| LinearSVC | O(m n) | No | Yes | No |
| SGDClassifier | O(m n) | Yes | Yes | No |
| SVC | $O(m^2 n)$ to $O(m^3 n)$ | No | Yes | Yes |

Table: Comparison of Scikit-Learn classes for SVM classification

## SVM Regression

- Try to fit as much instances as possible on the street.
- Width of street is controlled by $\epsilon$ epsilon.
- Larger $\epsilon$ leads to wider street.

# Chapter 6
# Decision Trees

## Topics discussed:

- Visualizing a decision tree.
- Make predictions with decision tree.
- CART training algorithm.
- Regularization.
- Limitations of decision tree.

## Visualizing a Decision Tree

1. You can visualize the trained Decision Tree by first using the export *_graphviz*(). method to output a graph definition file called *iris_tree.dot*
2. convert this .dot file to a variety of formats such as PDF or PNG using the dot command-line tool from the graphviz package.

# Chapter 6
# Decision Trees

Make predictions with decision tree:

**Samples:** Attribute that counts how many training instances it applies to.

**Value:** Attribute tells you how many training instances of each class this node applies to.

**Gini:** Attribute that measures the impurities.

### CART

Classification And Regression Tree (CART)

1. the algorithm first splits the training set in two subsets using a single feature k and a threshold $t_k$.
2. the algorithm keeps splitting recursively.
3. It stops recursing once it reaches the maximum depth or if it cannot find a split that will reduce impurity or if there is a stoppig condition from the hyperparameters.

# Chapter 6
# Decision Trees

## Regularization:

- To avoid overfitting the training data, you need to restrict the Decision Tree's freedom during training.
- Increasing *min** hyperparameters or reducing *max** hyperparameters will regularize the model.

## Limitations of decision trees

Limitations:

- Decision Trees love orthogonal decision boundaries, which makes them sensitive to training set rotation.
- They are very sensitive to small variations in the training data.

Thank you!