O'Reilly Hands-On Machine Learnign with Scikit-Learn and TensorFlow

Philip Tracton

November 2, 2018

Contents

1 Chapter 01 The Machine Learning Landscape

Machine Learning is the science and art of programming computers so they can *learn from data*

1.1 Definitions

- **training sets** are the examples the system uses to learn from.
- **training samples** are the samples or data in the training set
- **training data** is the new data used after training to see if it worked
- **data mining** is applying the techniques of machine learning to large amounts of data to discover patterns that were not immediately apparent
- **labels** training data fed to your algorithm that includes desired solutions
- **features** are an attribute and its value
- **agent** a reinforcement learning system it can observe environment, take actions, get rewards for good actions and penalties for bad ones
- **learning rate** how fast an algorithm adapts to changing data
- **similarity measure** is a method of seeing how close to samples are to each other.
- **utility function** is a measure of how good your functon is
- **cost function** is a measure of how bad your function is.
- **sampling noise** happens when there is too little data and you get non-representative data as chance
- **sampling bias** if the sampling method is flawed and leads to non-representative data.
- **Feature selection** is selecting the most useful feature of those available to train on
- **Feature Extraction** is combining one or more existing features into a single more useful feature (dimensionality reduction)

- **regularization** is constraining a model to make it simpler and reduce the risk of overfitting
- **hyperparameter** is a parameter of the learning algorithm, not the model, so it is not affected by training and must be set prior to training
- **generalization error** is the error rate on new cases. Done by evaluating model on the test set
- **cross validation** split the training set into complimentary subsets and each model is trained against a different combination of sets and validated against the remaining parts.
- **No Free Lunch** if you make no assumptions about the data there is no reason to prefer one model over others.

1.2 Concepts

1.2.1 Types of Machine Learning Systems

- Trained with human supervision
- Learn incrementally on the fly
- compare new data points to old data points and predict.

1. Supervised Learning
    - Used labelled training data
    - Typically used for classification tasks
    - Typically used for predicting target number. Given *features* it can go through a regression to predict new values.
    - Some Supervised Learning Algorithms in book
        - k-Neareset Neighbor
        - Linear Regression
        - Logisitic Regression
        - Support Vector Machines
        - Decision Trees and Random Forests
        - Neural Networks
2. Unsupervised Training

- Training data is unlabelled.
- Some important unsupervised learning algorithms
- Detect groups via clustering
- Reduce dimensionality to simplify data without loosing information
- Anomaly detection of finding outliers in data sets
- association rule learning is to dig into a large data set and discover interesting relations between attributes
- Visualization generate 2d or 3d representation of the data you feed it
  - Clustering
    * k-Means
    * Hierarchical Cluster Analysis
    * Expectation Maximization
  - Visualization and Dimensionality Reduction
    * Principal Component Analysis
    * Kernal PCA
    * Locally Linear Embedding
  - Association Rule Learning
    * Apriori
    * Eclat

3. Semi-Supervised Learning
   - partially labelled training data. Usually mostly unlabelled with some labelled data
   - Use a combination of supervised and unsupervised algorithms

4. Reinforcement Learning System
   - The agent (learning system) observes the environment and gets awards or penalties.
   - It must learn on its own the best strategy to maximize rewards and minimize penalties over time.

5. Batch Learning
   - System is incapable of learning over time and must be trained with with all available data

- This is called offline learning.

6. Online Learning

    - Incremental trainig by feeding it sequential data in small groups
    - Good for systems that receive a continuous flow of data
    - Can be used to train systems of huge data that do not fit into memory (out of core learning)
    - Incremental learning is a better name for this
    - bad data will cause system performance to decline over time
    - must manage learning rate, too fast will forget old information and too slow will be hard to adapt

1.2.2 Instance Based vs Model Based

- Good performance on training data is nice but true goal is good performance on new instances

1. Instance Based

    - The system learns examples by heart and generalizes to new cases using a similarity measure.

2. Model Based

    - Make a model from the examples and use that to make a prediction on new data samples.
    - Model selection can be a challenge.

1.2.3 Main Challenge of Machine Learning

1. Insufficient Quantity of Training Data

    - Need many thousands of examples to do this correctly.

2. Nonrepresentative Training Data

    - Model will behave based on training data. If it is not similar to production data, then the model will give poor results.
    - Be aware of sampling noise and sampling bias
    - Leads to inaccurate predictions

3. Poor Quality Data
    - If data is full of outliers, errors and noise, the algorithm will fail to detect underlying patterns
    - Worth time and effort of cleaning up data
        - May help to remove outliers
        - Fill in missing data? Fill in with what? Mean, Median, 0?
4. Irrelevant Features
    - Garbage In Garbage Out
    - This only works if you have enough relevant features and not too many irrelevant features.
    - Do feature selection
    - Do feature extration
5. Overfitting the Training Data
    - Model performs well on training data but fails to generalize on other data
    - This is over generalizing.
    - Happens when the model is too complex compared to the amount and noise of training data
    - Regularize your model
6. Underfitting the Training Data
    - Opposite of over fitting. Happens when you algorithm is too simple for the data
    - Fix with
        - More powerful algorithm
        - Better features
        - Reducing constraints

1.3 Testing and Validating

- Only way to tell if this works is to try on new cases
- Split your data into 2 set, training and testing.

- Common fix is to have a 3rd set of data, validation set.
- Process
  - Train many models an hyperparameters on the training data
  - Select the ones that perform the best for running with the validation set
  - Run a final test with the test data
- Use cross-validation

1.4 Exercises

1.4.1 How would you define Machine Learning?

- A method to train computers to perform better based on data or experience.

1.4.2 Can you name 4 types of problems where it shines?

- Problems with long lists of rules
- Complex problems with no good solutions by traditional methods
- Rapidly changing environments
- Getting insights into complex problems with a lot of data

1.4.3 What is a labelled training set?

- Data that includes the desired solutions

1.4.4 What are 2 most common supervised tasks?

- Classification
- Regression

1.4.5 Can you name 4 common unsupervised tasks?

- Clustering
- Visualization
- Dimensionality Reduction
- Association Rule Learning

1.4.6 What type of Machine Learning Algorithm would you use to allow a robot to walk in various unknown terrains?

- Reinforcement

1.4.7 What type of algorithm would you use to segment your customers into multiple groups?

- Unsupervised clustering

1.4.8 Would you frame the problem of spam detection as a supervised or unsupervised learning problem?

- Supervised

1.4.9 What is online learning?

- Incrementally and sequentially feeding small amounts of data to the algorithm. Good for systems with continuous data flow.

1.4.10 What is out of core learning?

- Learning from huge data sets that can not fit in the machine's memory, so you get it in pieces

1.4.11 What type of algorithm relies on a similarity measurement to make predicitions?

- Instance based learning find most similar instance and make predictions

1.4.12 What is the difference between a model parameter and a learning algorithm's hyperparameter?

- Hyperparameters are used to try to tune the various model's parameters to find optimal solutions

1.4.13 What do model based learning algorithms search for? What is the most common strategy they use to succeed?

1.4.14 How do they make predictions?

- Search for the best parameters so the model will generalize well when presented new data

1.4.15 Can you name 4 of the main challenges in Machine Learning?

- Insufficient Quantity of Data
- Nonrepresentative training data
- Poor quality data
- Irrelevant features
- Overfitting data
- underfitting data

1.4.16 If your model performs great on training data, but generalizes poorly to new instances what is happening?

1.4.17 Can you name 3 possible solutions?

- Simplify the model
- Gather more training data
- Reduce noise in training data

1.4.18 What is a testing set and why would you want to use it?

- Split your data into training and test. Training teaches the algorithm and test is used to show that it worked or not.

1.4.19 What is the purpose of a validation set?

- After using training data to train multiple algorithms, pick the best and try it on the validation set before using the test set

1.4.20 What can go wrong if you tune hyperparameters using the test set?

- You can overfit the test set

1.4.21 What is cross-validation and why would you prefer it to a validation set?

- Lets you compare models and hyperparameter settings without the need for separate validation sets

2 End to End Machine Learning Project

2.1 Definitions

- **pipeline** is a sequence of data processing components. Generally a series of asynchronous, self contained modules, consume a large block of data and create new results. Later another module does the same until we reach the end. This needs a lot of monitoring to make sure all is going well.

- **RMSE** is *Root Mean Square Error*. This is a typical performance measure for regression problems. Defined as $RMSE(\mathbf{X}, h) = \sqrt{\frac{1}{m}\Sigma_{i=1}^{m}(h(\mathbf{x}^{(i)} - y^{(i)}))^2}$. This will measure the standard deviation of the errors in predictions that the system makes.

- **Mean Absolute Error** is defined as $MAE(\mathbf{X}, h) = \frac{1}{m}\Sigma_{i=1}^{m}(h(\mathbf{x}^{(i)} - y^{(i)}))^2$

- **Data Snooping Bias** happens when estimating the error using the test set and you will be too optimistic.

2.2 Working with Real Data

- A lot of different source of data
  - Kaggle
  - Amazon Datasets
  - Data Portals
  - Reddit Datasets

2.3 Look at Big Picture

- This chapter is a project to build a model of CA Housing Prices.

2.3.1 Frame the Problem

- What is the goal of this model?

- What is business goal?

- What does final solution look like?

2.3.2 Select Performance Measure

- RMSE is the generally preferred performance measurement for regression work.
- Mean Absolute Error may prefer to use this if there are a lot of outliers.
- Both are ways of measuring the distance between two vectors. Various distance measurements or *norms* are possible.
  - Euclidian Norm
  - Manhattan Norm
  - The higher the norm index the more it focuses on large values and neglects small ones. This is why RMSE is more sensitive to outliers than MAE

2.4 Get the Data

- Hands On ML Data
- *export ML*$_{PATH}$*=<wherever you put this data>/ml*
- Virtual Environment. I think they are optional, but for a production set up it makes sense.
- Jupyter Notebooks
- Pandas
  - read\$_{csv}$() method reads the specified CSV file and returns a pandas data frame of the material
  - head() method to show first n rows of data frame
  - info() method to show concise summary of a data fame
  - describe() Generates descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN values.

```
import os
import pandas as pd
import matplotlib.pyplot as plt

HOUSING_CSV_PATH = "../handson-ml/datasets/housing/"
```

```python
HOUSING_CSV_FILE = "housing.csv"

def load_housing_data(housing_path=None):
    """
    In a very unsafe manner load the house csv file into a pandas data frame
    """
    csv_path = os.path.join(housing_path, HOUSING_CSV_FILE)
    return pd.read_csv(csv_path)

housingData = load_housing_data(HOUSING_CSV_PATH)
print(housingData.head())
print(housingData.info())
print(housingData.describe())
housingData.hist(bins=50, figsize=(20,15))
# plt.show()
# plt.savefig('../Notes/images/HousingHistogram.png', bbox_inches='tight')
```

2.4.1 Create a Test Set

-

2.5 Discover and Visualize Data to Gain Insights

2.6 Prepare the Data for Machine Learning

2.7 Select a Model and Train it

2.8 Fine Tune Your Model

2.9 Present Solutions

2.10 Launch, Monitor and Maintain

3 Classification

3.1 MNIST

3.2 Training a Binary Classifier

3.3 Performance Measures

- Measuring Accuracy Using Cross Validation
- Confusion Matrix

3.4 Precision and Recall

- Tradeoff

3.5 ROC Curve

3.6 Multiclass Classification

3.7 Error Analysis

3.8 Multilabel Classification

3.9 Multioutput Classification

3.10 Exercises