Companion to Machine Learning

Rohan Kumar

Contents

Introduction

0.1 What is Machine Learning

Machine Learning is the field of study that gives computers the ability to learn from data without being explicitly programmed. This is good for problems that require a lot of fine-tuning or for which using a traditional approach yields no good solution. Machine Learning's data dependency allows it to adapt to new data and gain insight for complex problems and large amounts of data.

0.2 Applications of Machine Learning

Machine Learning can be used for a range of tasks and can be seen used in:

- Analyzing images of products on a productino line to automaticall classify them (Convolutional Neural Net)
- Forecasting company revenue based on performance metrics (Regression or Neural Net)
- Automatically classifying news articles (NLP using Recurrent Neural Networks)
- Summarizing long documents automatically (Natural Language Processing)
- Building intelligent bot for a game (Reinforcement Learning)

0.3 Types of Machine Learning

Supervised Learning — Hands-On Machine Learning

In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels. (e.g determining if an email is spam would be trained a dataset of example emails labelled as spam or not spam.)

Some commonly used supervised learning algorithms are:

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural Networks

Unsupervised Learning — Hands-On Machine Learning

In unsupervised learning, the training data is unlabeled and the system tries to learn without guidance. The system will try and automatically draw inferences and conclusions about the data and group it as such. (e.g. having a lot of data about blog visitors. Using a clustering algorithm we can group and detect similar visitors).

Some important unsupervised learning algorithms are:

- Clustering
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis
- Anomaly detection and novelty detection
  - One-class SVM
  - Isolation Forest
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally Linear Embedding (LLE)
  - t-Distributed Stochastic Neighbor Embeedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat

Semisupervised Learning Hands-On Machine Learning

Labelling can be very time-consuming and costly, often there will be plenty of unlabelled and a few labelled instances. Algorithms that deal with data that is partially labeled is called semisupervised learning. A good example of this is Google Photos. Google clusters and groups your photos based on facial recognition (unsupervised) and then you can label one photo and it will be able to label every picture like that (supervised). Most semisupervised learning algorithms are combinations of unsupervised and supervised algorithms.

Reinforcement Learning Hands-On Machine Learning

Reinforcement Learning is a learning algorithm based on a reward system. The learning system, called an agent, can observe the environment, select and perform actions, and get rewards in return (or penalites in the form of negative rewards). It will then learn by itself what the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

Sources

Throughout this compendium, each piece of information will be formatted as such.

Name / Description of fact Source

Information about fact.

The location which currently contains "Source" could potentially be filled with a variety of sources. Here is how to find the source based off the shortened form.

- **Hands-On Machine Learning** refers to Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition by Aurélien Géron

1 Data Analysis

TODO:

- Data Model
- Overfitting/Underfitting
- Feature Normalization

2 Linear Regression

Formulation

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. Our main objective is to generate a line that minimizes the distance from the line to all of data points. This is essentially minimizing the error and maximizing our prediction accuracy.

Simple Regression

A simple two variable linear regression uses the slope-intercept form, where m and b are the variables our algorithm will try to "learn". x represents our input data and y represents the prediction.

$$y = mx + b$$

Multivariable Regression

Often times there are more than one feature in the data and we need a more complex multi-variable linear equation as our hypothesis. We can represent our hypothesis with the follow multi-viarable linear equation, where $\boldsymbol{w}$ are the weights and $\boldsymbol{x}$ is the input data.

$$\begin{aligned} h_{\boldsymbol{w}}(\boldsymbol{x}) &= w_0 x_0 + w_1 x_1 w_2 x_2 + ... + w_n x_n && (1) \\ &= \boldsymbol{w}^T \boldsymbol{x} && (2) \end{aligned}$$

Cost Function

To predict based on a dataset we first need to learn the weights that minimize the mean squared error (euclidean loss) of our hypothesis. We can define the following to be our cost function to minimize with m being the number of datapoints and i being the i^{th} training example.

$$J(\boldsymbol{w}) = \frac{1}{2m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}(\boldsymbol{x}^i) - \boldsymbol{y}^i)^2 \tag{3}$$

$$= \frac{1}{2m}(X\boldsymbol{w} - \boldsymbol{y})^T(X\boldsymbol{w} - \boldsymbol{y}) \tag{4}$$

Gradient Descent Solution

Now to solve for $\boldsymbol{w}$ we can use Gradient Descent and iteratively update $\boldsymbol{w}$ until it converges using the following:

$$\boldsymbol{w}_j := \boldsymbol{w}_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}(\boldsymbol{x}^i) - \boldsymbol{y}^i)$$

Normal Equation Solution

The closed form solution to the linear system in $\boldsymbol{w}$

$$\frac{\partial J\boldsymbol{w}}{\partial \boldsymbol{w}_j} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^T\boldsymbol{x}^i - \boldsymbol{y}^i)\boldsymbol{x}_j^i = 0$$

writing this as a linear system in w we get $A\boldsymbol{w} = b$ where

$$A = \sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{x}_n^T) \text{ and } b = \sum_{n=1}^{N}(\boldsymbol{x}_{\boldsymbol{n}}y_n)$$

so we can solve for $\boldsymbol{w} = \boldsymbol{A}^{-1}\boldsymbol{b}$ and get the following vectorized solution.

$$\boldsymbol{w} = (X^TX)^{-1}X^T\boldsymbol{y}$$

Index