# Index of Algorithms

Aagam Dalal

November 16, 2020

# Contents

# 1 Introduction

When I set out on this project I did not know what to expect. Was I going to have enough math knowledge? Were my programming skills good enough? The answer, in the end, was no on both accounts. But, that left plenty of room for growth, which I have done at every stage of this project. My goal in this document is to share what I have learned.

## 1.1 GitHub Repository

Access to the code generated during this process can be found on GitHub at `https://github.com/AaDalal/senior_sem_ai`. You may click "clone" if you would like to download some or all of it.

## 1.2 Getting Started

### 1.2.1 Coding

Python (`python.org`) is a versatile programming language that is commonly used in data science. Python has an extensive open-source development community that builds packages to extend its functionality. Anaconda (`https://docs.conda.io/projects/conda/en/latest/user-guide/install/`) is a multi-language package manager and environment software that is often used with Python. Anaconda comes with Anaconda Navigator, which is a graphical version of the Anaconda command-line interface. Built into Anaconda navigator's interface is access to JupyterLab. JupyterLab (`https://jupyter.org/`) is an interactive alternative to traditional command-line or script-based Python. It runs as a local webserver. Each Jupyter "notebook" is organized into cells of code and explanation, which makes it well suited for data science applications.

### 1.2.2 Research and Mathematics

Many of the concepts covered rely on background mathematical concepts at various levels of math. Khan Academy has an excellent series on multivariable calculus at `https://www.khanacademy.org/math/multivariable-calculus`. The YouTube channel 3blue1brown, run by Grant Sanderson, (`https://www.3blue1brown.com/`) provides simple explanations of much of the underlying math.

Google has an excellent program on Machine Learning through its machine learning crash course (`https://developers.google.com/machine-learning/crash-course`). StatQuest ((`statquest.org`)) with Josh Starmer is a YouTube

channel that provides simple but useful explanations of many AI topics. `towardsdatascience.org` features some excellent community-written articles on specific machine learning topics. Francois Chollet's *Deep Learning with Python* was an excellent specific resource for understanding deep learning and Keras. Stanford's cs229 (`online.stanford.edu/courses/cs229-machine-learning`) provides excellent in-depth mathematical descriptions of algorithms. MIT's OpenCourseWare courses on AI/ML (`https://ocw.mit.edu`) have similarly deep mathematical explanations included. As with any programming project, `stackoverflow.com` and `stackexchange.com` are important parts of working through bugs and understanding concepts.

These are just a few of the resources are used. You can find more in the GitHub repository (especially under books) or by looking at the citations in each notebook

### 1.2.3 Datasets

Many of the datasets I used were from Chollet's *Deep learning with Python*, *Introduction to Practical Statistics in R*, or `Kaggle.com`, which is a machine learning competition website.

### 1.2.4 Development Environment

Throughout the project, I used Conda to manage my Python packages. You can find the Conda environment at `https://github.com/AaDalal/senior_sem_ai/blob/master/conda_env.yml`. You can use the .yml file to recreate the environment I used by importing to Anaconda.

## 1.3 Using this Document

This document is broken up into categories. Within each category will be the specific topics that have been covered. The heading of each of those specific topics will link to the Jupyter notebook within the GitHub Repository

# 2 Exploratory Data Analysis

Data exploration is a vital part of machine learning. If the data is poorly understood, the techniques applied will not be appropriate, and the conclusions drawn from the data will also be poor.

## 2.1 Initial Data Analysis

Initial Data Analysis is the process of combing through data to find the inconsistencies within the data and correcting them. An example may be that data is mislabeled, missing, or in the wrong format. Initial Data Analysis makes sure that the data will be usable.

## 2.2 Statistical EDA

If initial data analysis is about making data usable, exploratory data analysis (EDA) is about making sure it is productive by getting more understanding of it. Statistical EDA is about understanding the feature distributions and some information about the bivariate relationships.

# 3 Clustering

Clustering is the process of sorting groups of spatially distributed variables into groups.

## 3.1 K Nearest Neighbor

K Nearest Neighbor (KNN) requires some data to already be clustered. A new data point is then clustered based on the classes of the K nearest points to it. KNNs need a lot of data compared to other clustering methods but produce more accurate results.

## 3.2 K Means

K Means is a recursive algorithm with a set number of clusters. The center of each cluster is initialized with random values. Then, at each step, each point is assigned to a cluster based on variance, and the center of the cluster is redefined to be the mean of all the points. K means is best used when the initial centers are as far apart as possible. With this goal in mind, KMeans++ was introduced. Overall, K means is a highly effective algorithm but it requires the number of clusters to be set.

## 3.3 DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Application with Noise. The longer name is indicative of its function: to use density to make clusters while still allowing some point to be categorized as noise. DBScan starts at an arbitrary point and, given that it has a certain number of points within a certain radius and all of the surrounding points to a new cluster. Then it repeats the process with all the surrounding points, then with all the points surrounding those, and so on until it terminates and re-initializes at a new point. This process can be summarized as following a chain of points until they terminate. The "certain number of points" is called minpts (for the minimum number of points). The "certain radius" is called eps (standing for $\epsilon$) and is typically visually estimated from a k-dist graph. The eps is set at the level where there is a slight valley in the k-dist graph before it changes slope.

# 4 Regression

Regression is very simply the idea of quantifying the relationship between two variables. These two variables usually an explanatory variable and a response variable, meaning that the explanatory variable can be used to explain the value of the response variable.

## 4.1 Least-Squares

Least-Squares regression is also sometimes called the sum of squared residuals (SSR) regression. This second more descriptive name reveals what is happening. The goal is to minimize the mathematical function $\sum(y_i - \hat{y})^2$. Because the residuals are squared, it is especially important to standardize the data by taking the z score for every point: $z = \frac{x-\mu}{\sigma}$ so that outlier points with high residuals do not dominate. Even with standardization, however, outlier points can be extremely influential. Despite this weakness, least squares regression is still the go-to for many real-world scenarios that depict linear or linearizable data.

# 5 Deep Learning

Deep learning describes neural networks that use more than just I/O layers. These layers in between, called hidden layers, are used to extract meaning out of data where it is difficult to find. This can be helpful, as they may not require heavy feature engineering, or harmful, as they overfit.

## 5.1 Multilayer Perceptron

The multilayer perceptron is foundational in machine learning. As the neuron is to the neural network, the perceptron is to the neural network. Each perceptron is a simple mathematical function consisting mainly of the dot product. But together can do any mathematical calculation. It has been theorized that just three layers of densely connected perceptrons can perform any mathematical transformation [CITATION NEEDED]. The method of training of the multilayer perceptron is the most challenging part to understand. It is helpful to think of the error of the network as the function to be minimized with respect to certain parameters of the neural network. Just like any minimization problem in calculus, you must take the first derivative. However, in this case, taking the first derivative means going backward through the neural network.

## 5.2 Recursive Neural Networks

Recursive Neural Networks (RNNs) are similar to any other neural network. However, instead of being strictly linear, where the output of each layer is used in the next, an RNN includes both the output of the previous layer and the original input. The result is that the final output can be more informed by the original context of the RNNs. When imagining RNN, it is useful to imagine a regular multilayer perceptron as a game of telephone. The RNN is a version of telephone where a syllable of the original word is told to each person. As you can imagine, the RNN version of telephone will result in a result closer aligned to the initial input. RNNs are, therefore, very useful, but can suffer from the vanishing gradient problem where optimizations become weaker as more layers are added to a neural network.

## 5.3 Convolutional Neural Network

A convolution is often defined as a curl or a twist [CITATION NEEDED]. Similarly, a convolutional neural network (CNN) is a twist on the traditional multilayer perceptron. CNNs are designed with visual data in mind. Like

an eye scanning across an image, the CNN first scans through chunks of the image. These chunks are then fed into a neural network where each neuron applies a transformation to the data. All the transformations are then grouped back together to form a transformed image with multiple different channels. The transformations to the image are trained to use spatial hierarchies to make identifications (Chollet 2018). Spatial hierarchies can be understood using Chollet's example of identifying an animal as a cat: first, you look at its ears and eyes to tell it is a land mammal and then using more subtle features like its claws to identify it as a cat. The transformations produced by convolution neurons would likely not be as intuitive to humans, but they would ideally identify spatial hierarchies in much the same way. The output of the convolutional neural network is usually fed to a traditional, densely connected multilayer perceptron to categorize the data after it has been transformed.

# 6 Misc

## 6.1 Naive Bayes

Think about making a prediction, given a certain set of information. Intuitively, you would use what you've seen in the past in addition to the data in front of you to make a decision. How could you codify this? Using Bayes' Theorem (1):

(1) $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$
(2) $P(Y|X) = Posterior$
(3) $P(Y) or P(Y|E) = Prior$
(4) $P(X|Y) = Likelihood$
(5) $P(D|E) = Normalization$

You have Prior beliefs, which is what you would assume without the data. Typically, this is the strict proportion of samples that have a certain category. You have the likelihood, which is the probability of observing the data given the observation. You have the Normalization, which is how likely you are to observe the data at all. You must divide this out because you want to prove that your prediction is having already been "given" the data. To be precise, $P(X|Y)$ might require that you already observed a case with the right mix of data. This might not be true when you have many features to your data which each has many options. Instead, you can naively assume that all of the features of the data are unrelated. The result is that the likelihood can be expressed as the simple multiple of the probabilities of all features.

In application, $Y$ is considered to be a class that a data-point $X$ can be grouped into. Therefore, $P(X|Y)$ means the probability of $X$ being a part of the class $Y$. The values of $P(X|Y)$ are compared for all the different classes. The one with the greatest probability is considered to be the class that $X$ is predicted to be in. Oftentimes, Naive Bayes will be used in Natural Language Processing Scenarios.

## 6.2 Support Vector Machines

Classification problems are commonly described in a geometric context as drawing a decision boundary between groups of points. Logically, the most optimal boundary is one that maximizes the distance to both classes. The analogy I use to understand is that of a two-lane road driving through the mountains (similar to what Patrick Winston at MIT uses). The line at the center that separates traffic moving in opposite directions is the decision boundary. The two lanes of the road are of equal width. The goal is to make the road as big as possible in between the mountains. The mountains

that are near the lanes of the road define how wide it can be. These closest "mountains" are called support vectors, and are what all of the math of the SVM revolves around.

The support vector classifier (SVC) codifies the idea of making the road as big as possible within the bounds of the support vectors using linear algebra and calculus. An SVC's boundaries are all hyperplanes, meaning they are straight. However, this poses a problem when the optimal decision boundary is not straight. The solution is to increase the dimensionality of the data until an SVC can be used. The process of increasing dimensionality and then applying the SVC is the work of the support vector machine. Central to this is something called the kernel trick, which allows the projection to higher dimensionality to be simulated without actually going through the computationally expensive projection.

## 6.3 Gradient Boosting Machines

Gradient Boosting Machines (GBM) are among the most common classifiers used in modern-day machine learning competitions. Their basis is a simple decision tree. In GBMs, multiple decision trees are combined to form what is called a decision forest. Each tree makes a "vote" as to what the classification should be. The gradient boosting part of the name refers to how each new decision tree reduces the error of the forest by adding decision criteria where there is the most error being produced.

Depending on the complexity and number of trees, GBMs can over-fit on the data. Even still, they are the go-to tool for classifications because of their great accuracy. Of note is the XGBoost version of GBM which has proved dominant due to its optimization, feature-set, and ease of use.