A Mathematician's Guide to Machine Learning

Sean Z. Roberson

University of Texas at San Antonio

September 17, 2024

Who Am I?

- Mathematician, operations researcher, (former) musician
- B.S. Mathematics, Texas A&M University - San Antonio (2015)
- M.S. Applied/Industrial Mathematics, University of Texas at San Antonio (2024)
- Also studied briefly at Texas A&M University

UTSA The University of Texas at San Antonio™

What is Machine Learning?

There are a few ways to think about machine learning:

- From *Mathematics for Machine Learning*, Deisenroth et. al.: "Machine learning is about designing algorithms that automatically extract valuable information from data."
- From IBM: "Machine learning (ML) is a branch of... computer science that focuses on the [use] of data and algorithms to enable AI to imitate the way that humans learn..."

But simply put: **It's how computers learn from past experiences.**

UTSA The University of Texas at San Antonio™

Chihuahua or Muffin?

Goals

At the end of this talk, you will...

- understand the basic mathematical framework of machine learning
- identify basic supervised and unsupervised learning tasks

Roadmap

1. The Mathematical Framework
2. Supervised Learning
3. Unsupervised Learning

What's Needed?

For this talk, we assume a basic knowledge of...

- differential calculus (take derivatives, find extrema)
- elementary probability

Linear algebra (beyond basic matrix computations) is nice to have, but will be mentioned on the fly. Other advanced topics, such as topology, functional analysis, and numerical analysis, will be left to dedicated sources.

The Objective (Function)

- Every machine learning starts with an objective.
- Mean-squared error, likelihood, and cross-entropy are common examples.
- The aim is to minimize (or maximize in some contexts) the objective subject to some constraints.

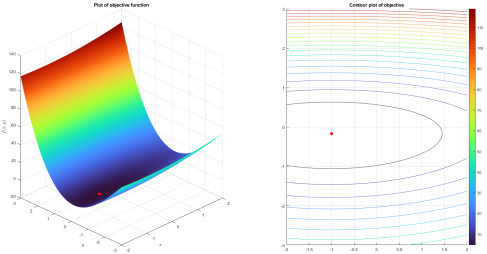


Figure: Example of an objective function.

Finding Minima

Elementary calculus gives a necessary condition for a minimum.

Necessary Condition for Minima

Suppose $f(x)$ admits a minimum at x^*. Then $f'(x^*) = 0$. In several variables, all derivatives must be zero, i.e., $\nabla f(x^*) = 0$.

But this isn't enough! We need to ensure we avoid points of inflection and saddle points.

Sufficient Condition for Minima

Suppose $f'(x) = 0$ and $f''(x) > 0$. Then f has a minimum at x. (The several variables case uses the Hessian matrix - the matrix of second derivatives.)

The issue is that finding the zeros of the derivative may involve nonlinear equations (sometimes many!) and the task needs to be solved numerically.

Minimization Algorithms

- Finding minima by iterative root-finders can be expensive, so line search methods are used.
- These include (stochastic) gradient descent methods.
- A gradient descent algorithm moves in the direction of largest descent - this is the direction where the derivative is largest.

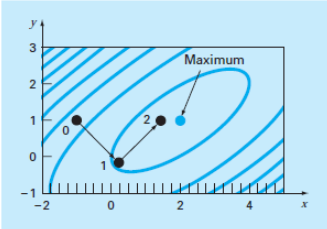


Figure: Illustration of gradient ascent. From *Numerical Methods for Engineers*, Chapra and Canale.

Which Algorithm?

- **Batch gradient descent** computes the gradient after the whole data set is processed. While stable, this can be very slow, especially for larger sets!
- **Stochastic gradient descent** is popular – the gradient is updated after every training example. This is good for larger sets, but the step sizes need to be tuned.
- **Adaptive gradient descent** methods are also popular, changing the step size progressively with the previous values of the gradient. While best suited for large models, sufficient computational resources need to be available.

The moral of the story: Consider the performance of your model and the size of your data set when choosing your optimization algorithm!

UTSA The University of Texas at San Antonio™

Roadmap

1. The Mathematical Framework
2. Supervised Learning
3. Unsupervised Learning

What Is Supervised Learning?

- A supervised learning task takes in a labeled data set and learns patterns in order to predict outcomes.
- Regression tasks aim to predict continuous outcomes, and classification predicts a category.
- The labels act as the "teacher!"

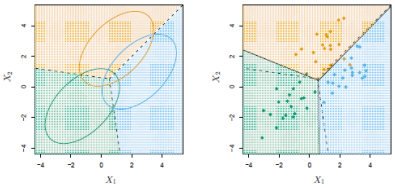


Figure: Example of a three-class problem. From *An Introduction to Statistical Learning with Python*.

Regression - A Simple Supervised Task

The simplest form of regression is as follows - given a collection of points (x_i, y_i), can we find a line of the form $y = \beta_0 + \beta_1 x$ that best fits through the data?

The word "best" here means to minimize the sum of squared residuals – the distance between the observation and the prediction. Thus, our objective is to minimize the function

$$J(\beta_0, \beta_1) = \sum_{i=1}^{N} (\beta_0 + \beta_1 x_i - y_i)^2.$$

We need to find β_0, β_1 (intercept and slope) that minimizes this function!

Solving the Regression Problem

The simple regression problem is easily solved through some calculus (steps omitted):

$$\beta_1 = \frac{\text{Cov}(X, Y)}{\text{Var}(X)}, \ \beta_0 = \bar{y} - \beta_1 \bar{x}$$

where $\text{Cov}(X, Y)$ is the covariance between the inputs $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_N)$ and $\text{Var}(X)$ is the variance of X.

In this case, there isn't much to learn - the problem has a tidy solution!

But with scikit-learn...

The approach in scikit-learn is better optimized for large (and possibly badly behaved matrices).

The computation of the inverse matrices needed is avoided by using either singular value decomposition (talk coming soon!) or a QR factorization.

With this approach, and by using a train-test split, the solution given by scikit-learn is close to the true solution given by the linear algebra.

UTSA
The University of Texas at San Antonio™

Visualizing Regression

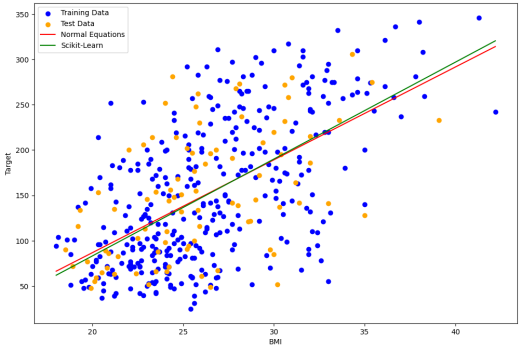
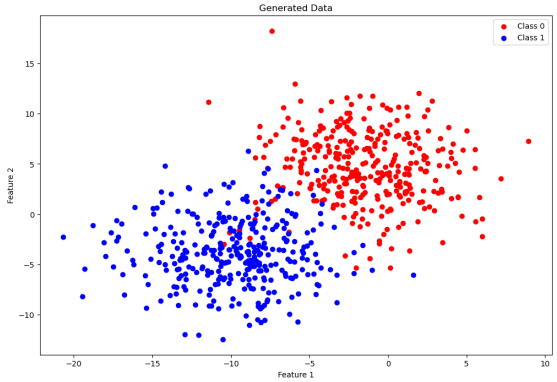


Figure: Plotting target treatment levels against BMI. From the diabetes set included in scikit-learn.

A Not-So-Tidy Problem: Classification

Instead of regression, let's consider the classification problem. Suppose you have N points in the plane, each of which is either in group A or group B. Can we find a way to easily identify which point belongs to which group?



Popular Classification Techniques

- Logistic regression (based on an odds ratio for binary classes)
- Support vector machines (attempts to linearly separate data)
- Decision trees (learns with yes/no questions)
- Nearest neighbors (predicts based on nearby data)

Same Data, Different Approaches

We demonstrate some of the more visual classification techniques below.

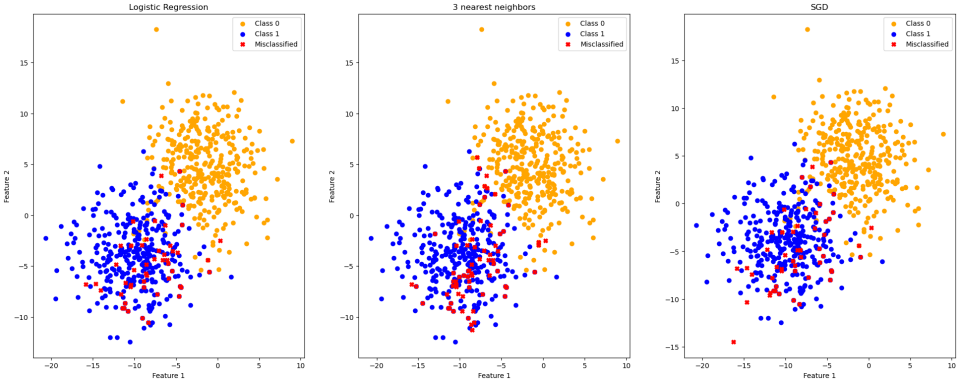


Figure: Three supervised algorithms.

Roadmap

Learning Without Labels

Unsupervised algorithms don't have labels given to them. Instead, they learn based on the similarities between the inputs. This is similar to:

- finding common features across the same or similar dog breeds
- spotting shapes in images to detect objects
- finding patterns in medical images to detect tumors

Unlike supervised learning, there is no human element - set it and forget it!

A Clustering Example

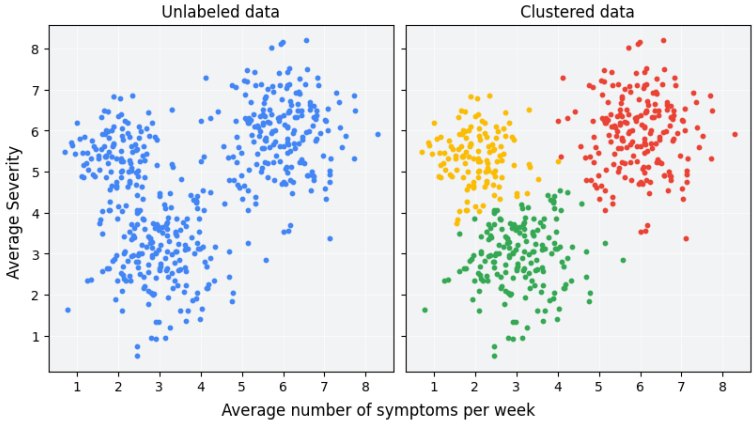


Figure: A clustering example. From Google for Developers.

Where Do We Go Now?

We can extend some of these methods to data sets with atypical geometries (see: topological data analysis and optimization on manifolds). The models can learn the geometry of the data.

Neural networks are constructed with the preceding ideas in mind - train a model, update parameters by minimization, and repeat. Constructing a good neural network, however, is an art!