Chapter 1

ML Introduction

September 2, 2021

Math/stats background

- Partial derivatives, e.g.:

$$\frac{\partial(x^3 + y^2 + 1)}{\partial x}$$

- Matrix and vector operations

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 8 + 3 \times 9 \\ 4 \times 7 + 5 \times 8 + 6 \times 9 \end{bmatrix} = \begin{bmatrix} 50 \\ 122 \end{bmatrix}$$

- Basic probability and statistics
  - Conditional probability
  - Normal distribution

Programming background

- Python
- NumPy
- Matplotlib

What is ML? Ask Wikipedia

Machine learning is a subfield of computer science (more particularly soft computing) that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. In 1959, Arthur Samuel defined machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed". Machine learning explores the study and construction of algorithms that can learn from and make predictions on data.

More definitions here

Applications of ML

- Image recognition
- Spam classification
- Web search engines
- Voice recognition
- Link to Quora

Three types of ML

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Yann LeCun explains supervised learning

A pattern recognition system is like a black box with a camera at one end, a green light and a red light on top, and a whole bunch of knobs on the front. The learning algorithm tries to adjust the knobs so that when, say, a dog is in front of the camera, the red light turns on, and when a car is put in front of the camera, the green light turns on. You show a dog to the machine. If the red light is bright, don't do anything. If it's dim, tweak the knobs so that the light gets brighter. If the green light turns on, tweak the knobs so that it gets dimmer. Then show a car, and tweak the knobs so that the red light get dimmer and the green light gets brighter. If you show many examples of the cars and dogs, and you keep adjusting the knobs just a little bit each time, eventually the machine will get the right answer every time.

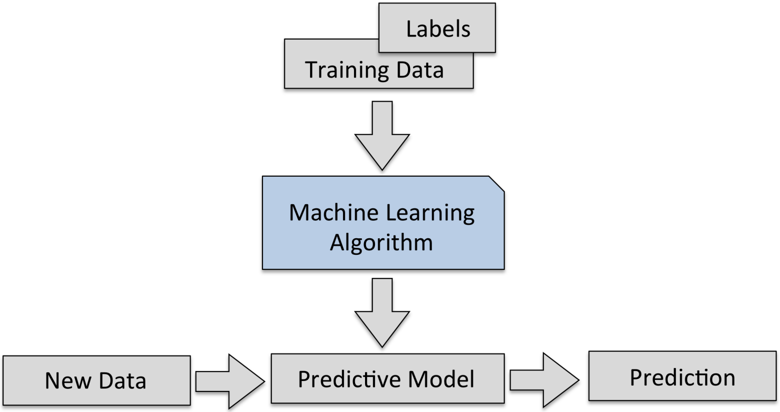
Scaling up

The interesting thing is that it may also correctly classify cars and dogs it has never seen before. The trick is to figure out in which direction to tweak each knob and by how much without actually fiddling with them. This involves computing a "gradient," which for each knob indicates how the light changes when the knob is tweaked.

Now, imagine a box with 500 million knobs, 1,000 light bulbs, and 10 million images to train it with. That's what a typical Deep Learning system is.

Source: IEEE Spectrum Interview

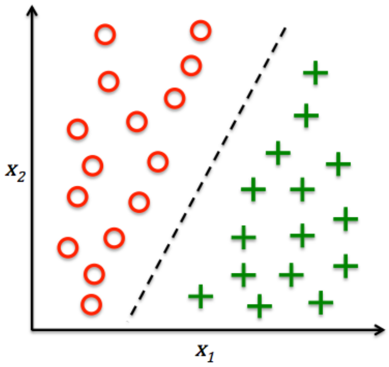
Supervised learning



- Predicting the future with supervised learning
- Classification vs. Regression

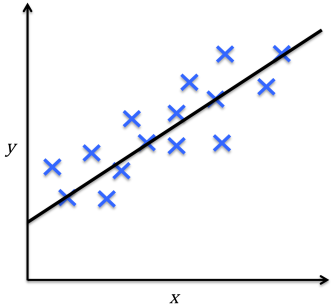
Classification

- Predict categorical class labels based on past observations
- Class labels are discrete unordered values
- Email spam classification example (binary)
- Handwritten digit classification example (multi-class)



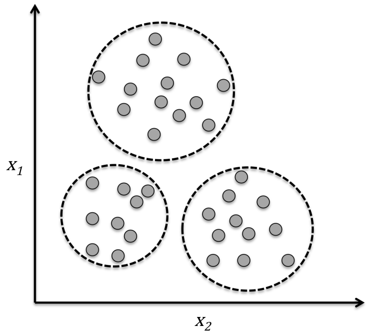
Regression

- Also a kind of supervised learning
- Prediction of continuous outcomes
- Predicting semester grades scores for students



Unsupervised learning

- Dealing with *unlabeled* data
- Cluster analysis
- Objects within a cluster share a degree of similarity



Unsupervised learing example

- Latent Dirichlet Allocation (LDA)
- Link to Wikipedia topics
- Wikipedia LDA entry
- Sara Palin topics

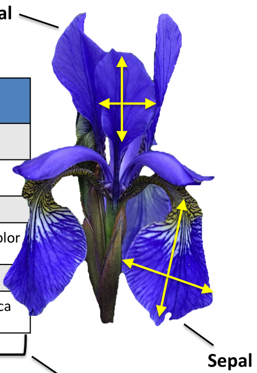
Iris dataset

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			…		
50	6.4	3.5	4.5	1.2	Versicolor
			…		
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)

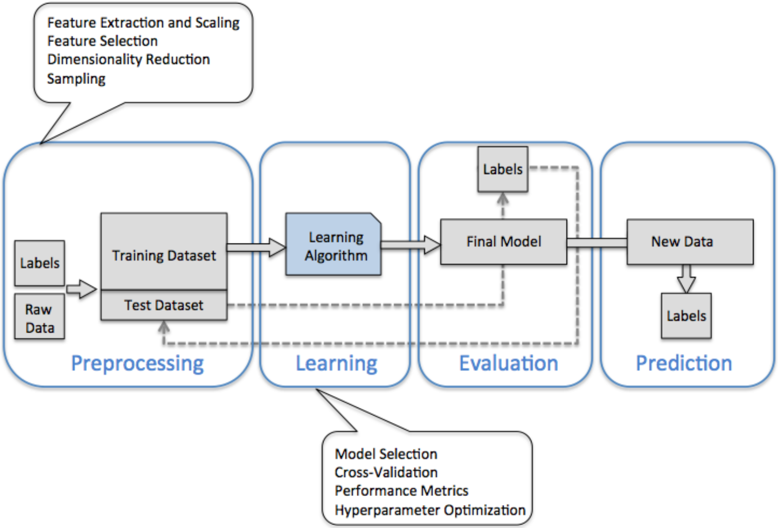


Basic terminology

- Measurements of 150 iris flowers (150 samples / 4 features)
- From 3 different species (Setosa, Versicolor, Virginica)
- Rows are samples and columns are features
- 150×4 matrix $\mathbf{X} \in \mathbb{R}^{150 \times 4}$:

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_4^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & \dots & x_4^{(150)} \end{bmatrix}$$

A roadmap for building ML systems



Model selection

No Free Lunch Theorem

Any two optimization algorithms are equivalent when their performance is averaged across all possible problems.

Example

A toy universe that exists for exactly two days and on each day contains exactly one object, a square or a triangle. The universe has exactly four possible histories:
(square, triangle)
(square, square)
(triangle, triangle)
(triangle, square)
Any prediction strategy that succeeds for history 2, by predicting a square on day 2 if there is a square on day 1, will fail on history 1, and vice versa. If all histories are equally likely, then any prediction strategy will score the same, with the same accuracy rate of 0.5.

Model selection

- What's the intuition behind NFL?
- Each classification algorithm makes assumptions
- Often we empirically determine what works best
- But how do we know what works best?
    - Classification accuracy
    - Train+Dev+Test split
    - Hyperparameter optimization (knobs of the model)

Python for ML

- Libraries for scientific computing such as NumPy and SciPy
- Performance of interpreted languages is inferior
- But NumPy and SciPy build upon lower level C and Fortran subroutines
- Scikit-learn library
- See page 13 for installation instructions (or just google)
- NumPy Slides