Machine Learning from Scratch

Project Update: Nov 28, 2017

Dan Johnston, Seattle

Project Overview

- Objective: manually code known machine learning algorithms
- Parameters:
  - Python, NumPy, SciPy only for implementation
  - Scikit-Learn for testing datasets and performance comparisons only
- Metrics
  - Model metrics (R2, RMSE, Accuracy, Recall, etc) are comparable to sklearn models
  - Reach goal: Runtime comparable to sklearn models
- Outcome:
  - Gain a deeper understanding for commonly used machine learning algorithms
  - Ability to relate the algorithms to a broad range of audiences

Current Status

Linear Regression

- Status: completed analytically (using linear algebra)
  - No regularization has been included.
- Boston dataset was used for comparison
- R^2 scores:
  - From Scratch: 0.740607742865
  - Sklearn: 0.740607742865
- Runtimes:
  - From Scratch: 153 µs ± 30.5 µs per loop
  - Sklearn: 438 µs ± 9.06 µs per loop

Logistic Regression

- Status: completed analytically (using linear algebra)
  - No regularization has been included
  - Binary classification only
  - Classification only, no class probabilities
- Breast Cancer dataset was used for comparison
- Accuracy scores:
  - From Scratch: 0.954305799649
  - Sklearn: 0.959578207381
- Runtimes:
  - From Scratch: 620 µs ± 31 µs per loop
  - Sklearn: 7.52 ms ± 313 µs per loop

K Neighbors Regression

- Status: completed with Euclidean distance only (using scipy.spatial.distance.cdist)
- Boston dataset was used for comparison
- R^2 scores:
  - From Scratch: 0.639665439953224
  - Sklearn: 0.639665439953224
- Runtimes:
  - From Scratch: 2.28 ms ± 79.3 µs per loop
  - Sklearn: 969 µs ± 29.6 µs per loop

K Neighbors Classifier

- Status: completed with Euclidean distance only (using scipy.spatial.distance.cdist)
  - Classification only, no class probabilities
- Breast Cancer dataset was used for comparison
- Accuracy scores:
  - From Scratch: 0.965034965034965
  - Sklearn: 0.965034965034965
- Runtimes:
  - From Scratch: 5.83 ms ± 68.2 µs per loop
  - Sklearn: 1.21 ms ± 47 µs per loop
  - Vote counting seems to be the bottleneck

What’s next?

More models!

Supervised Learning

- Naive Bayes
- Decision Trees
  - Ensembles?
- Stochastic Gradient Descent?
- Support Vector Machines?
- Linear Regression with Regularization?

Unsupervised Learning

- K Means clustering
  - Mini-batch too
- DB Scan clustering
- Mean Shift clustering?
- Principal Component Analysis?