

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 (R^2 , 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: $153 \mu\text{s} \pm 30.5 \mu\text{s}$ per loop
 - Sklearn: $438 \mu\text{s} \pm 9.06 \mu\text{s}$ per loop

Logistic Regression

- Status: completed analytically (using linear algebra)
 - No regularization has been included
 - Binary classification only
- Breast Cancer dataset was used for comparison
- Accuracy scores:
 - From Scratch: 0.965034965035
 - Sklearn: 0.979020979021
- Runtimes:
 - From Scratch: $563 \mu\text{s} \pm 55.9 \mu\text{s}$ per loop
 - Sklearn: $3.52 \text{ ms} \pm 86.3 \mu\text{s}$ 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 \pm 79.3 μ s per loop
 - Sklearn: 969 μ s \pm 29.6 μ s 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 \pm 68.2 μ s per loop
 - Sklearn: 1.21 ms \pm 47 μ s 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?