Gentle Introduction to Machine Learning with scikit-learn

Rob Zinkov

January 19th, 2012
Outline

1 Introduction

2 Machine Learning Basics

3 Scikit-Learn

4 Conclusion
What is the point of this talk?

- Get you playing around with Machine Learning techniques
- Get you excited about scikit-learn
Caveats

- This talk won’t change your life
- I won’t focus too much on techniques
- This talk is low on math
- This talk won’t make you an expert in scikit-learn
What is Machine Learning?

Machine Learning is the art of creating a compact explanation of the world using a large amount of data from the world.
Definitions

- **Model** the collection of parameters you are trying to fit
- **Data** what you are using to fit the model
- **Target** the value you are trying to predict with your model
- **Features** attributes of your data that will be used in prediction
- **Methods** algorithms that will use your data to fit a model

Note: Many methods are made to fit particular models
Which method should I use?

- Automatic Relevance Determination Regression (ARD)
- Bayesian Ridge Regression
- Lasso and Elastic Net
- Lasso path using LARS
- Lasso model selection: Cross-Validation / AIC / BIC
- Path with L1-Logistic Regression
- Ordinary Least Squares
- Orthogonal Matching Pursuit
- Polynomial interpolation
- Plot Ridge coefficients as a function of the regularization
- Plot multi-class SGD on the iris dataset
- SGD: Convex Loss Functions
Which method should I use?

- **Standard Answer:** Not really that important
- **Cynical Answer:** Whichever one performs the best
- **Less Cynical Answer:** The model that makes the most reasonable assumptions about your problem domain
But yes it's not the important question

Good Features are more important than Good Methods
Good Features is what counts
Simple Example Model

OK I don’t want to cheat you
Meet the Boston Housing Dataset
Boston House Prices dataset

Notes

Data Set Characteristics:

: Number of Instances: 506
: Number of Attributes: 13 numeric/categorical predictive
: Median Value (attribute 14) is usually the target
: Attribute Information (in order):
  - CRIM  per capita crime rate by town
  - ZN    proportion of residential land zoned for lots over 25,000 sq.ft.
  - INDUS  proportion of non-retail business acres per town
  - CHAS  Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  - NOX   nitric oxides concentration (parts per 10 million)
  - RM    average number of rooms per dwelling
  - AGE   proportion of owner-occupied units built prior to 1940
  - DIS   weighted distances to five Boston employment centres
  - RAD   index of accessibility to radial highways
  - TAX   full-value property-tax rate per $10,000
  - PRTATIO pupil-teacher ratio by town
  - B     1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - LSTAT % lower status of the population
  - MEDV  Median value of owner-occupied homes in $1000's

: Missing Attribute Values: None

: Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
http://archive.ics.uci.edu/ml/datasets/Housing

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

```python
In [21]: boston.data[1:100,1:6]
```
```
Out[21]:
array([[ 0. ,  7.07,  0. ,  0.469,  6.421,  0. ],
       [ 0. ,  7.07,  0. ,  0.469,  7.105,  0. ],
       [ 0. ,  2.18,  0.458,  0.524,  6.998,  0. ],
       [ 0. ,  2.18,  0.458,  0.524,  7.147,  0. ],
       [ 0. ,  2.18,  0.458,  0.524,  6.43 ,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  6.012,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  6.172,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  5.631,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  6.884,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  6.377,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  6.089,  0. ],
       [ 12.5,  7.87,  0. ,  0.524,  5.889,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.949,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.896,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.834,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.935,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.991,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.456,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.727,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.27 ,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.965,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.342,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.813,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.924,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.599,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.813,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.047,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.495,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.674,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.713,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.672,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.95 ,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  5.781,  0. ],
       [ 0. ,  8.14,  0.538,  0.538,  6.096,  0. ],
       [ 0. ,  5.96,  0.499,  0.499,  5.933,  0. ],
       [ 0. ,  5.96,  0.499,  0.499,  5.841,  0. ],
       [ 0. ,  5.96,  0.499,  0.499,  5.85 ,  0. ],
       [ 0. ,  5.96,  0.499,  0.499,  5.966,  0. ],
       [ 75,  2.05,  0.428,  0.428,  6.555,  0. ],
       [ 75,  2.05,  0.428,  0.428,  7.024,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  6.77 ,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  6.169,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  6.211,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  6.869,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  5.682,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  5.766,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  6.03 ,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  5.399,  0. ],
       [ 0. ,  6.91,  0.448,  0.448,  5.682,  0. ],
       [ 21,  5.64,  0.439,  0.439,  5.963,  0. ]])
```
This data plotted might resemble this
We assume these numbers can be linearly combined to predict housing price

\[ y(x, w) = w_0 + w_1 x_1 + \ldots + w_D x_D \]
Now this is how it's done

```python
from sklearn.linear_model import RidgeRegression
from sklearn import datasets

boston = datasets.load_boston()
X = boston.data
y = boston.target

clf = RidgeRegression()
clf.fit(X, y)
clf.predict(X)
```
What just happened?
Why Scikit-Learn?

- Weka is terrible
No really it is very bad

- Documentation is scattered
- Interfaces are terrible
- Code has well-known bugs
- Software is not actively maintained
- It’s Java code of the worst kind
Why Scikit-Learn?

- Weka is terrible
- Most libraries are just research code
If you’ve been there you know

- Documentation is non-existent (libsvm)
- Interfaces are idiosyncratic
- Software is unmaintained (libsvm, pybrain)
- Software is for educational purposes (nltk)
Why Scikit-Learn?

- Made on top of cython and scipy
- Fantastic community
Full of practitioners and researchers
Very Active

scikit-learn / Commit History

Jan 19, 2012
- Mutual Information docstring incorrectly said it was the adjusted mut... [commit]
  - Robert Layton authored about 6 hours ago

- Fix doctest.
  - mbiondel authored about 6 hours ago

- Merge branch 'warm_start'
  - mbiondel authored about 7 hours ago

- Revert "COSMIT refactor SGD code further"
  - larsms authored about 12 hours ago

- COSMIT refactor SGD code further
  - larsms authored about 15 hours ago

- Make sure order="C".
  - mbiondel authored about 19 hours ago

- Merge branch 'warm_start' of github.com:mblonde/scikit-learn into wa...
  - mbiondel authored about 19 hours ago

Jan 18, 2012
- Suppress deprecation warnings.
  - mbiondel authored 1 day ago

- Remove if statement.
  - mbiondel authored 1 day ago
Very pythonic
Getting it: You want the latest

```
pip install -U scikit-learn
```
A more realistic example

*That’s what she said*
Loading the example

```python
import numpy as np
y = np.concatenate((np.zeros(5796), np.ones(2091)))

DATADIR = "~/home/zv/custom_builds/twss-classifier/data"
data = itertools.chain(file(DATADIR+"/fmylife-parsed.txt"),
                        file(DATADIR+"/texts-from-last-night-parsed.txt"),
                        file(DATADIR+"/twss-stories-parsed.txt"))
```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data)
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB(0.01)
clf.fit(X,y)
text = "Something inappropriate"
if clf.predict_proba(vectorizer.transform(text)) > 0.995:
    print "TWSS"
Why this matters?

Machine Learning in Python isn’t a coincidence
Why this matters?

Python has a robust data ecosystem

- numpy
- scipy
- cython
- pandas
Why this matters?

- Python is what data scientists are using
- Python will become the center of the data science universe
Conclusions

Scikit-Learn is awesome, pythonic and fast
Conclusions

Now go make some cool!
References

- http://www.scikit-learn.org
- https://github.com/scikit-learn/scikit-learn
- http://nltk.org
- http://wit.io/posts/
  ruby-is-beautiful-but-im-moving-to-python
Questions?