Machine Learning with Python | scikit-learn review

Jorge F. Monardes
Pratt School of Engineering - Duke University

Objectives

This project is a simple review of **scikit-learn**, which is an open source machine-learning library for Python. In this project I review and explain the following:

- Guidance on how to install Python and all the libraries you need for scientific analyses.
- Brief explanation of what are the libraries and tools for scientific Python.
- Give you sources from where you can continue learning using Python as your scientific analysis programming language.
- Review classification methods seen in class, like:
  - Ordinary Least Square Regression (OLS).
  - Ridge Regression.
  - Logistic Regression.
  - Support Vector Machines.

Introduction

This project explains step by step how to install **scikit-learn** and all the libraries you need in your Mac, to start using Python as your scientific analysis programming language. The importance of Xcode and gfortran as *virtual machine compiler* is discussed and the need for *package managers* like Homebrew and Pip are explained. Additionally, instructions for properly install & use **NumPy** (fundamental package for scientific computing with Python), **SciPy** (library which provides efficient numerical routines for integration and optimization), and **scikit-learn** are given.

Figure 1: Python's libraries and tools

Materials (Software & Libraries)

The following software and libraries were required to complete this review:

1. Python 3.5 or later version
2. Xcode & gfortran
3. Homebrew and Pip
4. NumPy, SciPy, and scikit-learn
5. matplotlib
6. PyCharm
7. Data:
   - Leukemia gene expression [1]
   - scikit-learn standard datasets [2].

Extensive documentation and tutorials about scikit-learn can be found on it's website [2].

Methods

Linear and **Non-linear** classification methods were used for the analyses in this project.

- Ordinary Least Square Regression (OLS).
- Ridge Regression.
- Logistic Regression.
- Support Vector Machines.

Dataset of Leukemia gene expression was used for classification between the two Leukemia types: AML and ALL.
Moreover, an academic example of SVM was developed. The dataset used is the classic iris flower (setosa, versicolor, and virginica), considering 3 variables: sepal width & length, and petal length.

Important Result

Illustration and visual understanding of SVM with it's cloud regions of belonging in 3-dimensional space. Furthermore, the skills developed using Python as a tool for statistical science was personally rewarding.

Mathematical Section

The first classification method used an *Ordinary Least Squares* regression model.

$$\min_{\beta} \|X\beta - y\|_2^2 \tag{1}$$

Later on with *Ridge Regression* a penalty was imposed and the residual sum of squares was minimized.

$$\min_{\beta} \|X\beta - y\|_2^2 + \lambda\|\beta\|_2^2 \tag{2}$$

Also, *Logistic Regression* was part of the study.

$$\min_{\beta,c} \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n} \log(\exp(-y_i(X_i^T\beta + c)) + 1) \tag{3}$$

Finally, *Support Vector Machine* was used using another dataset to illustrate 3D classification.

$$\min_{w,b,\zeta} \frac{1}{2}w^T w + C\sum_{i=1}^{n} \zeta_i \tag{4}$$

Results

The following is a summary of the results obtained from the **Leukemia gene expression** dataset:

Regression	RSS	Var Score
OLS	39.21	-0.19
Ridge	31.36	0.05
Logistic	56.00	0.59

Table 1: Results with Leukemia gene expression dataset

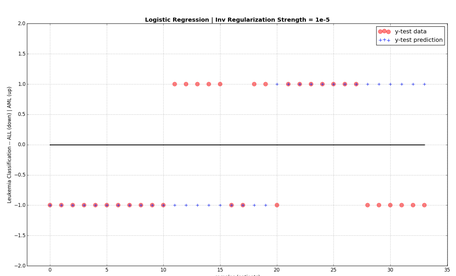

Figure 2: Logistic Regression | Leukemia (ALL / AML)

Results

Furthermore, below you can find 3D results using the **iris flower** dataset.

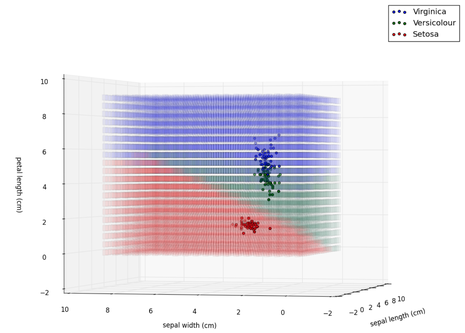

Figure 3: SVM with linear kernel | iris flower

Conclusion

This project helped me to strengthen my Python skills and gave me a practical understanding of some methods seen in class.

- The RSS (*Residual Sum of Squares*) could be deceiving. Always analyse with multiple statistics.
- Graphical analyses are fundamental for real understanding of the methods and results.

References

[1] Mukherjee, S. (2015, August 1). Leukemia gene expression. Retrieved November 30, 2015, from https://stat.duke.edu/ sayan/561/2015/homeworks/
[2] Scikit-learn: Machine learning in Python. (2013). Retrieved November 6, 2015, from http://scikit-learn.org/

Acknowledgements

I have to thank the scikit-learn community, INRIA, Google, and other sponsors for bringing together such an amazing kit-library for Machine Learning.

Contact Information

- Email: jorge.monardes@duke.edu
- LinkedIn: linkedin.com/in/jorgefmonardes