

Scikit Learn

G. Varoquaux

Neurospin + INRIA: statistical learning and inference for brain imaging

F. Pedregosa, A. Gramfort, V. Michel, J. VanderPlas, O. Grisel,
E. Duschenay, B. Thirion...



Project vision and goals

Machine learning for applications and non-experts

- Focus on ease of use
- Light and easy-to-install package
- A general-purpose high-level language: **Python**

... with high standards

- State of the art algorithms
- High-quality bindings: performance and fine control

Open source

- BSD licensed
- Community-driven

API and design

Design principles

- Minimize number of different object interfaces
- Build abstractions for recurrent usecases
- Simplicity, simplicity, simplicity
(no framework, no pipelines, no dataset objects)

Code sample

```
from scikits.learn import svm
classifier = svm.SVC()
classifier.fit(X_train, Y_train)
Y_test = classifier.predict(X_test)
```

API and design

- All objects

```
estimator.fit(X_train, Y_train)
```

- Classification, regression, clustering

```
Y_test = estimator.predict(X_test)
```

- Filters, dimension reduction, latent variables

```
X_new = estimator.transform(X_test)
```

- Predictive models, density estimation

```
test_score = estimator.score(X_test)
```

- One day: on-line estimation

```
estimator.refit(X_train, Y_train)
```

Main features and algorithms

Supervised learning

- SVM – high-quality LibSVM bindings
- Sparse regression: Lasso and Elastic Net
 - Coordinate descent and LARS
- Logistic Regression (L1 & L2) – LibLinear bindings
- K-Nearest Neighbors: – Ball-Tree
- ...

Unsupervised learning

- Gaussian mixtures models (work in progress)

To wrap up...

Scikit Learn:

- Machine learning without learning the machinery
- Focus on readability and simplicity of code:
 - optimize only the critical parts
 - code review
 - unit testing
 - coding standards

Well-suited for applications

- We use it on large datasets
- Building blocks for application-specific algorithms

Suggestions welcomed

<http://scikits-learn.sf.net>