FONDAZIONE BRUNO KESSLER

http://mpba.fbk.eu

mlpy
machine learning py

high-performance Python package for predictive modeling

Davide Albanese, Stefano Merler, Giuseppe Jurman,
Roberto Visintainer, Cesare Furlanello

FBK – MPBA Research Unit, Trento, Italy

NIPS Workshop on Machine Learning Open Source Software

12th December 2008

FONDAZIONE BRUNO KESSLER

Main Issues
(in developing a Open Source ML library)

Modularity: setting up a correct methodological workflow requires fulfilling a complex pipeline of basic tasks

Maintenance: rapid prototyping of new algorithms allows keeping the library updated to state-of-the-art

Reproducibility: the experiments should be repeatable, so every single step should be exactly replicable

Usability: researchers should be able to build their own methodological pipeline

Efficiency: computing time and memory usage are relevant in most of ML tasks

FONDAZIONE BRUNO KESSLER

Our Answer

python™

Dynamic object-oriented programming language

- very clear, readable syntax
- portable
- stable and mature

NumPy

Python module

- provides fast N-dimensional array manipulation
- basic linear algebra functions
- tools for integrating C/C++ code

C

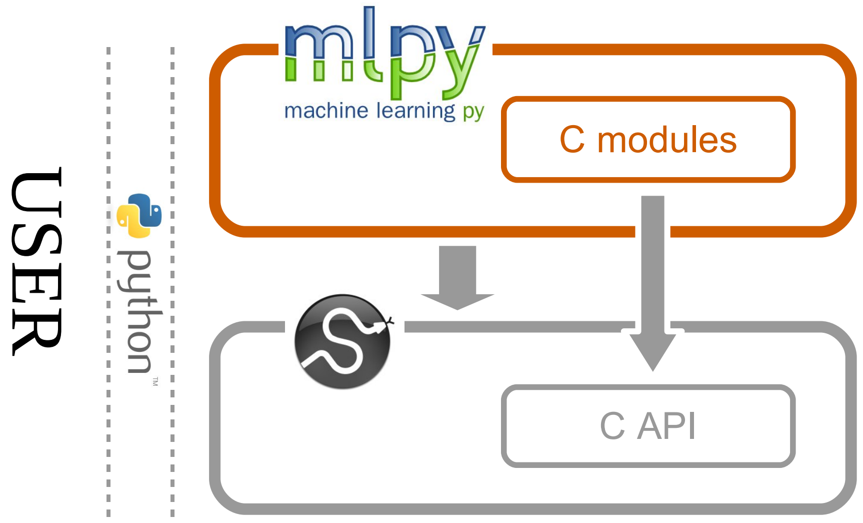
Well established and popular programming language

- efficiency
- code portability
- code reusing

mlpy v1.2.7 - Overview

Computationally efficient with low memory use

- internal ANSI C99 functions
- intensive use of the NumPy module



Multiplatform

- Unix and GNU/Linux
- MS Windows
- Mac OS X

Compact

- Source Code size: 464 KB
- ~3000 lines of ANSI C99 code
- ~2000 lines of Python code

Requirements

- libc
- Python >= 2.4
- NumPy >= 1.0.3

mlpy v1.2.7 - Structure

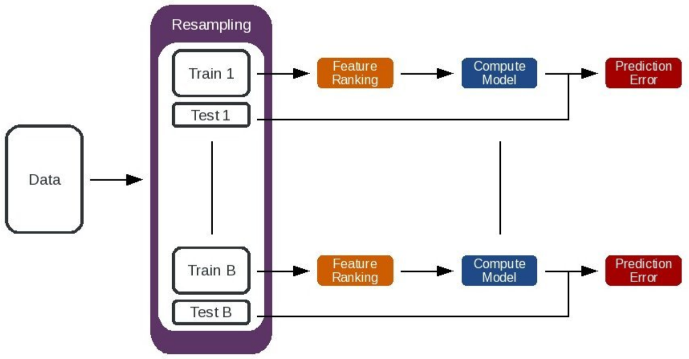
Classification	Feature Weighting	Feature Ranking	Feature List Analysis	Metric Functions	Resampling Methods
SVM	SVM	RFE	Stability	Error	TextBook
FDA	FDA	E-RFE	Extractions	Accuracy	MonteCarlo
SRDA	SRDA	BIS-RFE	Mean Position	Sensitivity	LOO CV
PDA	PDA	SQRT-RFE	Borda	Specificity	User Defined
NN	I-RELIEF	ONE-RFE		MCC	
	DWT	RFS		AUC	

Landscaping and Parameter Tuning Tools

svm-landscape	fda-landscape	pda-landscape
srda-landscape	irelief-sigma	nn-landscape

Provides high level procedures that support the design of rich ***Data Analysis Protocols*** (**DAPs**) for **predictive classification** and **feature selection**

Elective application field: **bioinformatics** on **high-throughput data**



Classification

Implemented Algorithms

- Support Vector Machines [Vapnik, 95]
  - Sequential Minimal Optimization (SMO) algorithm
  - Implemented in C
  - Four Kernels: Linear, Gaussian, Polynomial, Terminated Ramps [Merler and Jurman, 06]
- Nearest Neighbors [Cover and Hart, 67]
  - Implemented in C
- Discriminant Analysis
  - Fisher (KFDA) [Mika et al., 01]
  - Penalized (PDA) [Ghosh, 03]
  - Spectral Regression (SRDA) [Cai et al., 08]
  - Diagonal Linear (DLDA – mlpy v1.2.8) [Pique-Regi, 06]

- **`classifier(`*`params`*`)`**
  for classifier initialization.
- **`.compute(`*`x, y`*`)`**
  the method for the training phase computing the model. *x* stores the data (samples x features) and *y* collects the corresponding labels.
- **`.predict(`*`p`*`)`**
  the method for the testing phase predicting the model on a test-set. Test points are stored in *p*.
- ***`.realpred`***
  whenever possible it stores the real valued prediction.
- ***`._classifier__param`***
  internal classifier parameters are accessible.

FONDAZIONE
BRUNO KESSLER

Feature Weighting

Implemented Algorithms

- directly within SVM classifiers:
  - for all implemented kernels
- directly with DA:
  - Fisher (KFDA) – Cristianini method [Cristianini and Shawe-Taylor, 06]
  - Spectral Regression (SRDA)
  - Penalized (PDA)
  - Diagonal Linear (DLDA – mlpy v1.2.8)
- Iterative RELIEF (I-RELIEF) [Sun, 07]
- Discrete Wavelet Transform (DWT) [Subramani et al., 06]

- **`method(`*`params`*`)`** for feature weighting initialization.
- **`.weights(`*`x, y`*`)`** the method computing the feature score.
- ***`._method__param`*** internal parameters are accessible.

Feature Ranking

Implemented Algorithms

- Recursive Feature Elimination [Guyon et al., 02]
  - (Standard) RFE
  - Entropy-based RFE [Furlanello et al., 03]
  - Bisection RFE
  - Square-Root RFE
- Recursive/Sequential Forward Selection (R/S FS) [Louw and Steel, 06]
- One-step ranking

- `ranking(method, params)` for feature ranking initialization.
- `.compute(x, y, w)` the method computing the feature ranking. *w* is the feature weighting method. It returns the list of the ranked features.

Feature List Analysis

The ordered lists from the feature ranking experiments can be analyzed by:

`canberra(`*`lists, k`*`)`:

Canberra indicator on top-k positions [Jurman et al., 08]

`canberraq(`*`lists`*`)`(mlpy v1.2.8):

Canberra indicator on lists of different length

`borda(`*`lists, k`*`)`

- Extraction indicator
- Mean position indicator
- Optimal list on top-k sublists

HISTOIRE
DE
L'ACADÉMIE
ROYALE
DES SCIENCES.

ANNÉE M. DCCLXXXI.

Avec les Mémoires de Mathématique & de Phyſique, pour la même Année,
Tirés des Regiſtres de cette Académie.

A PARIS,
DE L'IMPRIMERIE ROYALE.

M. DCCLXXXIV.

JC de Borda, 1781

Jean Charles de Borda: Mémoire sur les élections au scrutin, in: Histoire de l'Académie Royale des Sciences. Année MDCCLXXXI, Paris, 1784
Photographically reproduced by Niels Peter Thomas, Technische Universität Darmstadt, 2003 (email: thomas@vwl.tu-darmstadt.de)

FONDAZIONE
BRUNO KESSLER

Metric functions

A set of different measure are available for the classifier performance assessment:

- Error
  - $err=(fp+fn)/ts$
  - $errp=fp/ap$
  - $errn=fn/an$
- Accuracy
  - $acc=(tp+tn)/ts$
- Sensitivity and Specificity
  - $sens=tp/ap$
  - $spec=tn/an$
- Matthews Correlation Coefficient (MCC)
  - $MCC=((tp\,tn)-(fp\,fn))/\sqrt{(tp+fn)(tp+fp)(tn+fn)(tn+fp)}$
- Area Under the ROC Curve (AUC)

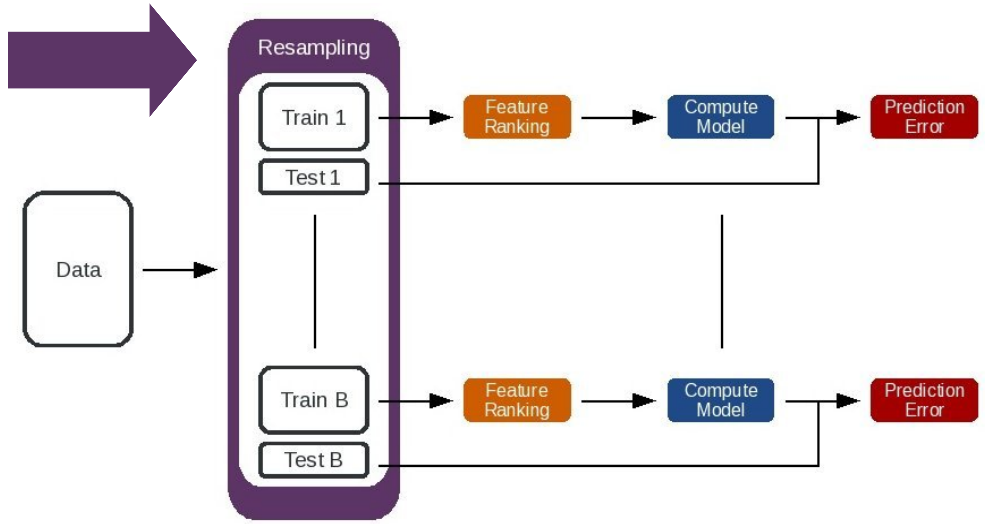
The Confusion Matrix		
ts	ap	an
pp	tp	fp
pn	fn	tn

Variability assessed by Bootstrap Confidence Intervals

Resampling Methods

A few sampling procedures available with focus on replicability:

- Textbook (k-fold) cross validation
- Monte-Carlo cross validation
- Leave-one-out cross validation
- User-defined train/test



- **`Method(`*`params`*`)`** returns a list of tuples which contain the sample indexes for each replicate.For example:

```
  training   test
[([2,4,5,6],[0,1,3]),
 ([0,1,5,6],[2,3,4]),
 ([0,1,2,3],[4,5,6]),
 ([1,2,3,4],[0,5,6]),
 ([0,2,4,6],[1,3,5]),
 ([0,1,2,5],[3,4,6])]
```

- **`StratMethod(`*`params`*`)`** the *Strat* prefix indicates that stratification over labels is available

Landscaping and Parameters Tuning Tools

The package includes executable scripts to be used *off-the-shelf* for landscaping and parameter tuning tasks. These scripts implement a basic DAP.

- `svm-landscape` (regularizer)
- `srda-landscape` (alpha parameter)
- `fda-landscape` (regularizer)
- `pda-landscape` (regressions steps)
- `nn-landscape`
- `irelief-sigma` (sigma parameter)

User can choose the resampling method, range and number of steps

Error, MCC and Canberra Distance are retrieved for each step

FONDAZIONE
BRUNO KESSLER

Notes

- mlpy is used by FBK-MPBA Research Unit for the MAQC-II project led by US FDA
- Runs on HPC facilities, Linux cluster at FBK and European Grid for E-sciencE (EGEE)
- mlpy is now used on datasets of **10^5 samples** and tested for up to **10^6 features**:
  - Copy Number Variation (CNVs)
  - Single Nucleotide Polymorphism (SNP)
  - Gene Expression (Microarray)
  - Proteomic (Mass Spectra)
- Partially supported by AIRC-IFOM
- Licensed under the GNU General Public License (GPL) version 3
- Homepage: https://mlpy.fbk.eu