Introduction to pySPACE workflows (https://github.com/pyspace)

a Signal Processing and Classification Environment written in Python

M M Krell, S Straube, A Seeland, H Wöhrle, J Teiwes, J H Metzen, E A Kirchner, F Kirchner

DFKI Bremen & University of Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de
pySPACE: Computation of Multiple Workflows

- ...with applications in robotics and brain-computer interfaces
- ...with simple configuration and automatic processing of empirical evaluations
- ...on feature vector and time series datasets
- ...where configuration requires no programming (YAML used)
- ...with execution in a distributed manner (embarrassingly parallel)
- ...intuitive structure
- ...choosing from more than 100 signal processing and classification algorithms (additionally interfaces to other libraries)
pySPACE: Computation of Multiple Workflows

- ...with **applications** in robotics and brain-computer interfaces
- ...with **simple configuration** and **automatic processing** of empirical evaluations
- ...on feature vector and time series datasets
- ...where configuration requires no programming (YAML used)
- ...with execution in a distributed manner (embarrassingly parallel)
- ...intuitive structure
- ...choosing from more than 100 signal processing and classification algorithms (additionally interfaces to other libraries)
... with **applications** in robotics and brain-computer interfaces

... with **simple configuration** and **automatic processing** of empirical evaluations

... on feature vector and time series datasets

... where configuration requires no programming (YAML used)

... with execution in a distributed manner (embarrassingly parallel)

... intuitive **structure**

... choosing from more than 100 signal processing and classification algorithms (additionally interfaces to other libraries)
Robotic application in the Project VirGo

Predict sensor values

- e.g. gyro, temperature, battery load

⇒ detect and react to unexpected events

Methods for predicting upcoming sensor readings are developed

pySPACE is used to:

- process different datasets,
- compare/evaluate different regression algorithms,
- and tune their parameters (e.g. nodes in the hidden layers of a Multilayer perceptron)
other Applications

- evaluation and comparison of
  - ...sensor selection algorithms (on EEG data) [2]
  - ...dimensionality reduction algorithms (ICA, PCA, xDAWN, PiSF, CSP) [3, 4, 5, 13]
  - ...classifiers (BRMM, online classifiers, ...) [6, 11, 14, 15]
- Brain-Computer Interfaces (movement prediction, interaction error detection, detection of warning perception) [1, 7, 8, 9, 10, 16, 17]
- soil detection
- parallelization of robot simulations
- classify iterative closest point (ICP) matches for good and bad localization
other Applications

- evaluation and comparison of
  - ... sensor selection algorithms (on EEG data) [2]
  - ... dimensionality reduction algorithms (ICA, PCA, xDAWN, PiSF, CSP) [3, 4, 5, 13]
  - ... classifiers (BRMM, online classifiers, ...) [6, 11, 14, 15]

- Brain-Computer Interfaces (movement prediction, interaction error detection, detection of warning perception) [1, 7, 8, 9, 10, 16, 17]

- soil detection
- parallelization of robot simulations
- classify iterative closest point (ICP) matches for good and bad localization
other Applications

- evaluation and comparison of
  - ... sensor selection algorithms (on EEG data) [2]
  - ... dimensionality reduction algorithms (ICA, PCA, xDAWN, PiSF, CSP) [3, 4, 5, 13]
  - ... classifiers (BRMM, online classifiers, ...) [6, 11, 14, 15]
- Brain-Computer Interfaces (movement prediction, interaction error detection, detection of warning perception) [1, 7, 8, 9, 10, 16, 17]
- soil detection
- parallelization of robot simulations
- classify iterative closest point (ICP) matches for good and bad localization
Short Facts

- medium sized framework (> 40000 lines of code)
- developed and tested on Mac OS X and Linux
- 5 years old (open source since August 2013)
- core developer team of 3-5 people and approx. 10 in total
- open source software (GPL, https://github.com/pyspace)
- extensive documentation: http://pyspace.github.io/pyspace/
- paper about pySPACE published yesterday:
  Mario Michael Krell, Sirko Straube, Anett Seeland, Hendrik Wöhrle, Johannes Teiwes, Jan Hendrik Metzen, Elsa Andrea Kirchner, and Frank Kirchner. pySPACE - A Signal Processing and Classification Environment in Python. Frontiers in Neuroinformatics, 7(40), 2013
Short Facts

- medium sized framework (> 40000 lines of code)
- developed and tested on Mac OS X and Linux
- 5 years old (open source since August 2013)
- core developer team of 3-5 people and approx. 10 in total
- open source software (GPL, https://github.com/pyspace)
- extensive documentation: http://pyspace.github.io/pyspace/
- paper about pySPACE published yesterday:
  Mario Michael Krell, Sirko Straube, Anett Seeland, Hendrik Wöhrle, Johannes Teiwes, Jan Hendrik Metzen, Elsa Andrea Kirchner, and Frank Kirchner. pySPACE - A Signal Processing and Classification Environment in Python. *Frontiers in Neuroinformatics*, 7(40), 2013
How to install and use pySPACE

1. installation (very simple, see tutorial)
2. prepare your data for pySPACE
3. decide and define the processing file
4. potentially modify your config file
5. start software
prepare your data: Input Formats

- feature vector: csv, arff
- time series segments: csv
- time series stream: csv, EDF2, .set (EEGLAB), .eeg (BrainProducts GmbH)
1. installation

2. **prepare your data**
   
   dataset description of banana dataset (metadata.yaml)
   
   ```yaml
   storage_format: [csvUnnamed, real]
   type: FEATURE_VECTOR
   file_name: banana_data.csv
   label_column: 1
   ...
   ```

3. decide and define the processing file

4. potentially modify your config file

5. start software
installation
prepare your data
decide and define the processing file (examples/bench.yaml)

```yaml
  type: node_chain
  input_path: "example_summary"
  runs: 3
  node_chain:
    - node: FeatureVectorSourceNode
    - node: TrainTestSplitter
      parameters:
        train_ratio: 0.4
    - node: __Normalization__
    - node: 2SVM
      parameters:
        complexity: __C__
    - node: PerformanceSinkNode

parameter_ranges:
  __C__: [0.01, 0.1, 1]
  __Normalization__: [GaussianFeatureNormalization, EuclideanFeatureNormalization]
```

potentially modify your config file
How to install and use pySPACE

1. **installation**

2. **prepare your data for pySPACE**

3. **decide and define the processing file (bench.yaml)**

4. **potentially modify your config file** (config.yaml)
   
   ```
   storage: ~/pySPACEcenter/storage
   spec_dir: ~/pySPACEcenter/specs
   console_log_level : logging.WARNING
   file_log_level : logging.INFO
   python_path:
       - /home/user/pySPACE/external/libsvm/python/
   ...
   ```

5. **start software**
1. installation
2. prepare your data for pySPACE
3. decide and define the processing file (bench.yaml)
4. potentially modify your config file (config.yaml)
5. **start software**
   go to pySPACEcenter on the command line and type:
   ```
   ./launch.py -o examples/bench.yaml --mcore
   ```
Parallelization

- single-core: — — serial
- multi-core: — — mcore
- cluster (common storage system needed): — — loadl
- possibility to add new modes: — — cloud

- online and offline mode
- no interprocess communication (restricted to embarrassingly parallel)
- shared file system required
Parallelization

- single-core: — — *serial*
- multi-core: — — *mcore*
- cluster (common storage system needed): — — *loadl*
- possibility to add new modes: — — *cloud*

Design decisions to enable parallelization:

- online and offline mode
- no interprocess communication (restricted to embarrassingly parallel)
- shared file system required
Concepts and Features

General Structure Concept

operation

node

operation chain

node chain

operation

node chain

node

dataset

node

FIR Filter

sub-sampling

SVM

summary

offline

在线

offline + online

summary

offline

offline

offline + online

dataset

summary

offline + online

offline

offline

offline + online

dataset

summary

offline + online

offline

Offline + online
More than 100 algorithms

- More than 100 algorithms
- Spatial Filter
- CSP
- xDAWN
- PCA
- ICA
- FDA
- Filters
- FFT
- IIR
- FIR
- TKEO
- Decimation
- Resample
- Window Function
- Normalizations
- detrend
- z-score
- baseline
- Avg. EEG
- Scatter
- Raw Data
- Instance Selection
- Others
- Debug
- Visualization
- Spectrum
- Histogram
- Train-Test Splitter
- Postprocessing
- Cross-validation
- Feature Selection
- Feature Normalization
- Naive Bayes
- Linear Fit
- Grid Search
- Pattern Search
- Coherence
- Linear Fit
- Fusion
- Grid Search
- Feature Generator
- Sub-Chain
- Pattern Search
- LIBSVM
- Naive Bayes
- Scikit-learn Wrapper
- LDA
- LIBLINEAR
- Classification
- QDA
- Random
- SOR SVM
- Gating Functions
- Debug
- Instance Selection
- Others
- Type Conversion
- Performance
- Score Mapping
- gaussian
- histogram
- precision
- weighted
- ridge
- regression
- label voting
- probability
- voting
- Debug
More than 100 algorithms

Modularity concept of node chain based on Modular toolkit for Data Processing (MDP)!
Here new algorithms/libraries can be integrated/interfaced!
Conclusion

- pySPACE automatizes the signal processing and classification workflow.
- automatic parallel execution of other evaluations (WEKA, Reinf. Learning with MMLF http://mmlf.sourceforge.net/)
- intuitive configuration without scripting (YAML based) ⇒ useable by non-programmers
- possibility to integrate other algorithms/libraries

future steps
- more algorithms and interfaces to other libraries
- more data types (e.g. pictures, videos)
- more applications (e.g. clustering, regression)
- installation suite
- ...
Conclusion

- pySPACE automatizes the signal processing and classification workflow.
- automatic parallel execution of other evaluations (WEKA, Reinf. Learning with MMLF http://mmlf.sourceforge.net/)
- intuitive configuration without scripting (YAML based) ⇒ useable by non-programmers
- possibility to integrate other algorithms/libraries

**Future steps**
- more algorithms and interfaces to other libraries
- more data types (e.g. pictures, videos)
- more applications (e.g. clustering, regression)
- installation suite
- ...
Thank you for your attention!

Do you have questions?

funding statement: Supported by the national aeronautics and space research centre of the Federal Republic of Germany (DLR) by grants of the Federal Ministry of Economics and Technology on the basis of a decision by the German Bundestag under the grant numbers 50RA1113 and 50RA1114 in the project VirGo4.


Sensor Selection

![Graph showing balanced accuracy vs. number of EEG electrodes](image)

- balanced accuracy
- number of EEG electrodes
- standard caps
- various methods (All, SSNR_VS, SSNR_AS, 1SVM, 2SVM, Performance, PCA, xDAWN, CSP, Standard caps)

M M Krell

pjSPACE (https://github.com/pyspace)

November 22, 2013 3 / 10
Balanced Relative Margin Machine 1/3

Error (%)
equi.
to SVM
variants
equivalent
to RFDA
variants
1–norm L1–BRMM
1–norm L1–RMM
2–norm L1–BRMM
2–norm L1–RMM
2–norm L2–BRMM
2–norm L2–RMM

Range $R$
0
5
10
15
20
25
30
35
40

M M Krell
pySPACE (https://github.com/pyspace) November 22, 2013 4 / 10
Balanced Relative Margin Machine 2/3

Individual Classification Problems

Range $R$

Relative Error to SVM (%)
Online classifier evaluation
Segmentation analysis
Installation

required dependencies:

- Python2.7
- YAML
- NumPy
- SciPy

optional dependencies:

- matplotlib (visualizations)
- scikit-learn (classifiers)
- PyQt4 (GUIs)
- LIBSVM, LIBLINEAR, MDP, ... (algorithm interfaces)

download (git clone https://github.com/pyspace/pyspace.git)

python setup.py ⇒ configuration folder in home directory
Installation

required dependencies:

- Python2.7
- YAML
- NumPy
- SciPy

optional dependencies:

- matplotlib (visualizations)
- scikit-learn (classifiers)
- PyQt4 (GUIs)
- LIBSVM, LIBLINEAR, MDP, . . . (algorithm interfaces)

download (git clone https://github.com/pyspace/pyspace.git)

python setup.py ⇒ configuration folder in home directory
Installation

required dependencies:
- Python2.7
- YAML
- NumPy
- SciPy

optional dependencies:
- matplotlib (visualizations)
- scikit-learn (classifiers)
- PyQt4 (GUIs)
- LIBSVM, LIBLINEAR, MDP, … (algorithm interfaces)

download (git clone https://github.com/pyspace/pyspace.git)

python setup.py ⇒ configuration folder in home directory
Why use pySPACE instead of other libraries or software?

- no scripting ⇒ usable for neuroscientists
- automatic parallel processing (e.g. on cluster)
- other libraries can be integrated
- a lot of available algorithms
- no separation between preprocessing, classification, parameter optimization, and evaluation
- easy exchange of processing schemes
- real open source
- working on cluster