将軍

Sören Sonnenburg[†], Gunnar Rätsch[♮], Fabio De Bona[♮]

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

MAX-PLANCK-GESELLSCHAFT

[†] Fraunhofer FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
[♮] Friedrich Miescher Laboratory of the Max Planck Society, Spemannstr. 39, 72076 Tübingen, Germany

Machine Learning Toolbox SHOGUN features algorithms:

- to learn 2-class classification and regression problems
- to train hidden markov models
- toolbox's focus is on kernel methods esp. Support Vector Machines (SVMs) for computational biology
- also implements a number of linear methods like Linear Discriminant Analysis (LDA), Linear Programming Machine (LPM), (Kernel) Perceptrons

Features

- SHOGUN interfaces to Matlab™, Octave and Python and R
- provides generic SVM object interfacing to several different SVM implementations, among them the state-of-the-art LibSVM and SVM^{light}
- SVMs can be trained using a variety of common kernels (Linear, Polynomial, Gaussian) and recent String Kernels (TOP, Fisher, Locality Improved, Spectrum, Weighted DegreeKernel (with shifts))
- kernels can be combined; weighting can be learned using Multiple Kernel Learning.
- input feature-objects can be dense, sparse or strings and of type int/short/double/char; can be converted into different feature types.
- multiprocessor parallelization $\Rightarrow$ able train on **10 million** examples

. . . and many more. . .

Generic Demo:

- Support Vector Classification
  - Task: separate 2 clouds of gaussian distributed points in 2D
  - octave, R, python
- Support Vector Regression
  - Task: learn a sine function
  - octave, R, python
- Hidden Markov Model
  - Task: 3 loaded dice are drawn 1000 times, find out when which dice was drawn
  - octave, R, python

Bioinformatics Demo:

- Position Independent (e.g. Tissue Classification using Promotor Region)

```
AAACAAAAACGTAACTAATCTTTTAGAGAGAACGTTTCAACCATTTTGAG
AAGATTAACTCATCACAGATTTCATTACATACAGATATAATTCAAAAATT
CACTCCCCAAATCAACGATATTTAAAAATCACTAACACATCCGTCTGTGC
```

  - Task: separate DNA strings, '-' class random ACGT, '+' class contains 'AAAAA' motif

- Position Dependent (e.g. Splice Site Classification)

```
AAACAAATAAGTAACTAATCTTTTAAGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAACATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAATTCACTAACACATCCGTCTGTGCC
```

  - Task: separate DNA strings, '-' class random ACGT, '+' class 'AA' in the middle

- Mixture Position Dependent/Independent (e.g. Promoter Classification)

```
AAACAAATAAGTAACTAATCTTTTAAAGAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTCATTAAATACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTAAATTTCACTAACACATCCGTCTGTGC
```

  - Task: separate DNA strings, '-' class random 'ACGT', '+' class 'AAA' in the middle shifted ± 15

Drawbacks and Future

Now:

- Interface to R,octave,matlab,python using same/very similar syntax

```
sg("set_features", "TRAIN", traindat)
sg("set_labels", "TRAIN", trainlab)
sg("send_command", "set_kernel GAUSSIAN REAL 40 1")
sg("send_command", "init_kernel TRAIN")
sg("send_command", "new_svm LIGHT")
sg("send_command", "c 10.0")
sg("send_command", "svm_train")
sg("set_features", "TEST", testdat)
sg("send_command", "init_kernel TEST")
out=sg("svm_classify")
```

```
sg('set_features', 'TRAIN', traindat)
sg('set_labels', 'TRAIN', trainlab)
sg('send_command', 'set_kernel GAUSSIAN REAL 40 1')
sg('send_command', 'init_kernel TRAIN')
sg('send_command', 'new_svm LIGHT')
sg('send_command', 'c 10.0')
sg('send_command', 'svm_train')
sg('set_features', 'TEST', testdat)
sg('send_command', 'init_kernel TEST')
out=sg('svm_classify')
```

- No "objects," i.e. not possible with > 1 SVM classifier

Future:

- Focus on a new python object oriented interface (C++ classes are directly wrapped via swig)
- $\Rightarrow$ much cleaner: Demo

Summary

- SHOGUN is a large scale machine learning toolbox
  $\Rightarrow$ able to train on **10 million** examples
- unified SVM framework + many string kernels suitable for comp. biology
- Algorithms: HMM, LDA, LPM, Perceptron, SVM, SVR + many kernels
- Interfaces to matlab,octave,python,R

Contribute + get support for all interfaces automagically!

Help us!

- Documentation
- Examples
- Testing
- Test Suite

Source Code is freely available under the GPLv2.

`http://www.fml.tuebingen.mpg.de/raetsch/projects/shogun`