The SHOGUN Machine Learning Toolbox

(and its interfaces)

Sören Sonnenburg[1,2], Gunnar Rätsch[2],Sebastian Henschel[2],Christian Widmer[2],Jonas Behr[2],Alexander Zien[2],Fabio de Bona[2],Alexander Binder[1],Christian Gehl[1], and Vojtech Franc[3]

[1] Berlin Institute of Technology, Germany
[2] Friedrich Miescher Laboratory, Max Planck Society, Germany
[3] Center for Machine Perception, Czech Republic

Outline

1. Introduction
2. Features
3. Applications and Demo

SHOGUN Machine Learning Toolbox - Overview I

History

1999	Initiated by S. Sonnenburg and G. Rätsch (SHOGUN)
2006	First public release (June)
2008	used in 3rd party code (PyVMPA)
Now	Several other contributors mostly from Berlin, Tübingen
	Debian, Ubuntu, MacPorts packaged, $>$ 1000 installations

Unified (large-scale) learning for various feature types and settings

Machine Learning Methods Overview

- Regression (Kernel Ridge Regression, SVR)
- Distributions (Hidden Markov models...)
- Performance Measures
- Clustering
- **Classification**

SHOGUN Machine Learning Toolbox - Overview II

Focus: Large-Scale Learning with...

- 15 implementations of Support Vector Machines solvers
- 35 kernels (Focus on string-kernels for Bioinformatics)
- Multiple Kernel Learning
- Linear Methods

Implementation and Interfaces

- Implemented in C++ ($> 130,000$ lines of code)
- **Interfaces:** libshogun, python, octave, R, matlab, cmdline
- **Over 600 examples**
- Doxygen documentation
- **Testsuite** ensuring that obvious bugs do not slip through

Feature Representations

Input Features

- Dense Vectors/Matrices (SimpleFeatures)
  - `uint8_t`
  - ⋮
  - float64_t
- Sparse Vectors/Matrices (SparseFeatures)
  - `uint8_t`
  - ⋮
  - float64_t
- Variable Length Vectors/Matrices (StringFeatures)
  - `uint8_t`
  - ⋮
  - float64_t

⇒ **loading and saving as hdf5, ascii, binary, svmlight**

Interfaces

Interface Types

- Static Interfaces (single object of each type only)
- Modular Interfaces (really object oriented, SWIG based)

Support for all Feature Types

- Dense, Sparse, Strings
- Possible by defining generic get/set functions, e.g.

```
void set_int(int32_t scalar);
void set_real(float64_t scalar);
void set_bool(bool scalar);
void set_vector(float64_t* vector, int32_t len);
void set_matrix(float64_t* m, int32_t rws, int32_t cls);
...
```

$\Rightarrow$ set/get functions for access from python, R, octave, matlab

The Eierlegendewollmilchsau^{TM} Interface

Embed Interface A from Interface B

- possible to run python code from octave
- possible to run octave code from python
- possible to run r code from python
- ...

Demo: Use matplotlib to plot functions from octave.

Unique Features of SHOGUN I

Input Features

- possible to stack together features of arbitrary types (sparse, dense, string) via CombinedFeatures and DotFeatures
- chains of "preprocessors" (e.g. substracting the mean) can be attached to each feature object (on-the-fly pre-processing)

Kernels

- working with custom pre-computed kernels.
- possible to stack together kernels via CombinedKerneli (weighted linear combination of a number of sub-kernels, not necessarily working on the same domain)
- kernel weighting can be learned using MKL
- Methods (e.g., SVMs) can be trained using unified interface

Unique Features of SHOGUN II

Large Scale

- multiprocessor parallelization (training with up to 10 million examples and kernels)
- implements COFFIN framework (dynamic feature / example generation; training on 200,000,000 dimensions and 50,000,000 examples)

Community Integration

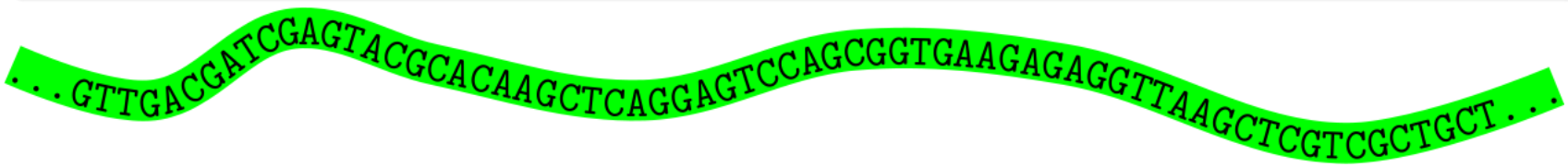
- Documentation available, many many examples
- There is a Debian Package, MacOSX
- Mailing-List, open SVN repository

. . . and many more. . .

Application

Genomic Signals

- Transcription Start (Sonnenburg et al., 2006)
- Acceptor Splice Site (Sonnenburg et al., 2007)
- Donor Splice Site (Sonnenburg et al., 2007)
- Alternative Splicing (Rätsch et al., 2005)
- Transsplicing (Schweikert et al., 2009)
- Translation Initiation (Sonnenburg et al., 2008)



Genefinding

- Splice form recognition - mSplicer (Rätsch et al. 2008)
- Genefinding - mGene (Schweikert et al., 2009)

Demo

Support Vector Classification

- Task: separate 2 clouds of gaussian distributed points in 2D
- Task: detect genomic signal

Support Vector Regression

- Task: learn a sine function

Hidden Markov Model

- Task: 3 loaded dice are drawn 1000 times, find out when which dice was drawn

Clustering

- Task: find clustering of 3 clouds of gaussian distributed points in 2D

Summary

SHOGUN Machine Learning Toolbox

- Unified framework, for various interfaces
- Applicable to huge datasets (>**50 million** examples)
- Algorithms: HMM, LDA, LPM, Perceptron, SVM, SVR + many kernels, ...

Documentation, Examples, Source Code

- Implementation `http://www.shogun-toolbox.org`
- Documentation `http://www.shogun-toolbox.org/doc`

We need your help:

- Documentation, Examples, Testing, Extensions

To appear in June in JMLR 2010 MLOSS track