The SHOGUN Machine Learning Toolbox
(and its R interface)

Sören Sonnenburg\textsuperscript{1,2}, Gunnar Rätsch\textsuperscript{2}, Sebastian Henschel\textsuperscript{2}, Christian Widmer\textsuperscript{2}, Jonas Behr\textsuperscript{2}, Alexander Zien\textsuperscript{2}, Fabio De Bona\textsuperscript{2}, Alexander Binder\textsuperscript{1}, Christian Gehl\textsuperscript{1}, and Vojtech Franc\textsuperscript{3}

\textsuperscript{1} Berlin Institute of Technology, Germany
\textsuperscript{2} Friedrich Miescher Laboratory, Max Planck Society, Germany
\textsuperscript{3} Center for Machine Perception, Czech Republic
Outline

1. Introduction
2. Features
3. Code Example
4. Summary
What can you do with the SHOGUN Machine Learning Toolbox [6]?

- **Types of problems:**
  - Clustering (no labels)
  - **Classification** (binary labels)
  - Regression (real valued labels)
  - Structured Output Learning (structured labels)

- Main focus is on **Support Vector Machines** (SVMs)
- Also implements a number of other ML methods like
  - Hidden Markov Models (HMMs)
  - Linear Discriminant Analysis (LDA)
  - Kernel Perceptrons
Support Vector Machine

- Given: Points \( x_i \in \mathcal{X} \ (i = 1, \ldots, N) \) with labels \( y_i \in \{-1, +1\} \)
- Task: Find hyperplane that maximizes margin

Decision function \( f(x) = w \cdot x + b \)
SVM with Kernels

SVM decision function in kernel feature space:

\[
f(x) = \sum_{i=1}^{N} y_i \alpha_i \Phi(x) \cdot \Phi(x_i) + b = k(x, x_i)
\]  

Training: Find parameters \( \alpha \)

Corresponds to solving quadratic optimization problem (QP)
Large-Scale SVM Implementations

- Different SVM solvers employ different strategies
- Provides generic interface to 11 SVM solvers
- Established implementations for solving SVMs with kernels
  - LibSVM
  - SVM$^{\text{light}}$
- More recent developments: Fast linear SVM solvers
  - LibLinear
  - SvmOCAS [1]
- Support of Multi-Threading

⇒ We have trained SVMs with up to 50 million training examples
Large Scale Computations

- Training time vs sample size
Large Scale Computations

- Training time vs sample size

![Graph showing training time vs sample size]

- 12k
- 1 min
Large Scale Computations

- Training time vs sample size

![Graph showing training time vs sample size](image)
Large Scale Computations

- Training time vs sample size

![Graph showing training time vs sample size.](image)
Various Kernel Functions

- Kernels for real-valued data

(a) Linear
(b) Polynomial
(c) Gaussian

⇒ What if my data looked like...
Various Kernel Functions

- Kernels for real-valued data

(d) Linear
(e) Polynomial
(f) Gaussian

⇒ What if my data looked like...
...this?!
Various Kernel Functions

- **String Kernels**
  - Applications in Bioinformatics [3, 5, 7], Intrusion Detection
  - Idea of Weighted Degree String Kernel
    \[
    k(s_1, s_2) = w_7 + w_1 + w_2 + w_2 + w_3
    \]

- **Heterogeneous Data Sources**
  - CombinedKernel class to construct kernel from weighted linear combination of subkernels
    \[
    K(x, z) = \sum_{i=1}^{M} \beta_i \cdot K_i(x, z)
    \]
  - \(\beta_i\) can be learned using Multiple Kernel Learning [4, 2]
Interoperability

- Supports many programming languages
  - Core written in C++ (> 130,000 lines of code)
  - R-bindings using SWIG (Simple Wrapper Interface Generator)
  - Additional bindings: Python, Matlab, Octave
  - More to come, e.g. Java

- Supports many data formats
  - SVM\textsuperscript{light}, LibSVM, CSV
  - HDF5

- Community Integration
  - Documentation available, many many examples (> 600)
  - Source code is freely available
  - There is a Debian Package, MacOSX
  - Mailing-List, public SVN repository (read-only)
  - Part of MLOSS.org
Simple Code Example

Simple code example: SVM Training

```r
# given: features, labels, test as R-data structures
lab <- Labels(labels)
train <- RealFeatures(features)
gk <- GaussianKernel(train, train, 1.0)
svm <- LibSVM(10.0, gk, lab)
svm$train()
out <- svm$predict(test)
```

- It's easy to train & predict
- Generic interface to many solvers (e.g. LibSVM → SVMLight)
- SVM accepts any kernel (e.g. GaussianKernel → PolyKernel)
When is SHOGUN for you?

- You want to work with SVMs (11 solvers to choose from)
- You want to work with Kernels (35 different kernels)
  ⇒ Esp.: String Kernels / combinations of Kernels
- You have large scale computations to do (up to 50 million)
- You use one of the following languages:
  R, Python, octave/MATLAB, C++
- Community matters: mloss.org, mldata.org
Thank you!

Thank you for your attention!!

For more information, visit:

- Implementation http://www.shogun-toolbox.org
- More machine learning software http://mloss.org
- Machine Learning Data http://mldata.org
References I

V. Franc and S. Sonnenburg.
Optimized cutting plane algorithm for large-scale risk minimization.

Efficient and accurate lp-norm multiple kernel learning.

mGene: Accurate SVM-based gene finding with an application to nematode genomes.

Large scale multiple kernel learning.
S. Sonnenburg, A. Zien, and G. Rätsch.


The SHOGUN machine learning toolbox.

*Journal of Machine Learning Research, 2010.*
(accepted).

C. Widmer, J. Leiva, Y. Altun, and G. Raetsch.
Leveraging Sequence Classification by Taxonomy-based Multitask Learning.