Outline

- Introduction
- Supervised learning with Shark
- Roundtrip through all the rest
- Conclusions
Introduction

What is Shark?
Introduction

What is Shark?

- machine learning library
Introduction

What is Shark?

- machine learning library
- C++
What is Shark?

- machine learning library
- C++
- modular
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL

Tobias Glasmachers

The Shark Machine Learning Library
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
  - pure machine learning, just algorithms
What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
  - pure machine learning, just algorithms ⇒ no own GUI
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
  - pure machine learning, just algorithms ⇒ no own GUI
  - completely self-contained:
    - flexible arrays
    - random number generator
    - linear algebra
    - ... other tools
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
  - pure machine learning, just algorithms ⇒ no own GUI
  - completely self-contained:
    - flexible arrays
    - random number generator
    - linear algebra
    - ... other tools
⇒ no dependencies!
Introduction

What is Shark?

- machine learning library
- C++
- modular
- free: GPL
- portable
  - OS-independent: Linux, MacOS, Windows
  - pure machine learning, just algorithms ⇒ no own GUI
  - completely self-contained:
    - flexible arrays
    - random number generator
    - linear algebra
    - ... other tools
  ⇒ no dependencies!
- growing, actively maintained
Introduction

What is Shark?

- well-documented

The Shark Machine Learning Library

Main Library Documentation

ReClMa
This library serves as a toolbox for Regression and Classification Methods. It provides different models for regression and classification, in particular kernel-based algorithms (e.g., SVMs, Gaussian Processes) and neural networks. For the adaptation of model parameters, ReClMa offers several gradient-based algorithms. To achieve high flexibility, ReClMa has a modular structure, in which model, error functional, and optimizer can freely be combined.

EALib
The EALib is a library providing Evolutionary Algorithms (in particular evolution strategies and genetic algorithms) and related techniques.

MOO-EALib
The MOO-EALib extends the EALib by providing various evolutionary Multi-Objective Optimization algorithms.

Tools Documentation

Mixture
Library for the representation and optimization of mixture density models.

Tobias Glasmachers

The Shark Machine Learning Library
Introduction

What is Shark?

- well-documented
- more than 60 example programs

... 

// create the SVM for prediction
RBFKernel k(gamma);
SVM svm(&k, false);

// create a training scheme and
// an optimizer for learning
C_SVM Csvm(&svm, C, C);
SVM_Optimizer SVMopt;
SVMopt.init(Csvm);

// train the SVM
SVMopt.optimize(svm, x, y);

...
Introduction

What is Shark?

- well-documented
- more than 60 example programs
- more than 20 tutorials

The Travelling Salesman Problem

This is an advanced example showing how to implement a tailored evolutionary algorithm using the EALib.

An exemplary problem: The travelling salesman problem

The travelling salesman problem is a combinatorial optimization task. A salesman is supposed to visit n cities. Each travelling connection is associated with a cost (i.e. the time for the trip). The problem is to find the cheapest round-route that visits each city exactly once and returns to the starting point.

The figure shows the example used in this tutorial with 10 cities.
Outline

- Introduction
- Supervised learning with Shark
- Roundtrip through all the rest
- Conclusions
Supervised learning with Shark

Learning ⇔ Optimization

Tobias Glasmachers

The Shark Machine Learning Library
Supervised learning with Shark

## Learning ⇔ Optimization

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>#parameter</td>
</tr>
<tr>
<td>+model()</td>
</tr>
<tr>
<td>+modelDerivative()</td>
</tr>
<tr>
<td>+getParameter()</td>
</tr>
<tr>
<td>+setParameter()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ErrorFunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>+error()</td>
</tr>
<tr>
<td>+errorDerivative()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>+init()</td>
</tr>
<tr>
<td>+optimize()</td>
</tr>
</tbody>
</table>
Learning $\Leftrightarrow$ Optimization

- **Model**
  - `#parameter`
  - `+model()`
  - `+modelDerivative()`
  - `+getParameter()`
  - `+setParameter()`

- **ErrorFunction**
  - `+error()`
  - `+errorDerivative()`

- **Optimizer**
  - `+init()`
  - `+optimize()`

- **Composite Model**
  - **RNNNet**
  - **KernelFunction**
  - **SVM**

Tobias Glasmachers | The Shark Machine Learning Library
Supervised learning with Shark

Learning ⇔ Optimization

- **Model**
  - `#parameter`
  - `+model()`
  - `+modelDerivative()`
  - `+getParameter()`
  - `+setParameter()`

- **ErrorFunction**
  - `+error()`
  - `+errorDerivative()`

- **Optimizer**
  - `+init()`
  - `+optimize()`

- **MeanSquaredError**
- **ZeroOneLoss**

Tobias Glasmachers
The Shark Machine Learning Library
Supervised learning with Shark

Learning ⇔ Optimization

Model
#parameter
+model()
+modelDerivative()
+getParameter()
+setParameter()

ErrorFunction
+error()
+errorDerivative()

Optimizer
+init()
+optimize()

BFGS
CMA-ES

Rprop
SVM-Optimizer

...
Supervised learning with Shark

Learning ⇔ Optimization

Model
- #parameter
- +model()
- +modelDerivative()
- +getParameter()
- +setParameter()

ErrorFunction
- +error()
- +errorDerivative()

Optimizer
- +init()
- +optimize()

BFGS

CMA-ES

Rprop

SVM-Optimizer

Very fast quadratic program solvers for all types of SVMs!

Tobias Glasmachers
The Shark Machine Learning Library
Supervised learning with Shark

Learning ⇔ Optimization

- Communication only through top level interfaces
Supervised learning with Shark

Learning ⇔ Optimization

- Communication only through top level interfaces
- Apply any combination of Model, ErrorFunction, and Optimizer
Application: Training a Support Vector Machine
Application: Training a Support Vector Machine

**L2-SVM**
- #gamma
- #C

**RBFKernel**
- #gamma

**SVM**
- #alpha
- #b
Supervised learning with Shark

Application: Training a Support Vector Machine

Diagram:
- L2-SVM
  - #gamma
  - #C
- ZeroOneLoss
- SVM-Optimizer
- RBFKernel
  - #gamma
- SVM
  - #alpha
  - #b
Supervised learning with Shark

Application: Training a Support Vector Machine

- **L2-SVM**
  - #gamma
  - #C

- **ZeroOneLoss**

- **SVM-Optimizer**

- **CrossValidationError**

- **GridSearch**

- **RBFKernel**
  - #gamma

- **SVM**
  - #alpha
  - #b

Tobias Glasmachers  The Shark Machine Learning Library
Supervised learning with Shark

Application: Training a Support Vector Machine

- L2-SVM
  - #gamma
  - #C

- ZeroOneLoss

- SVM-Optimizer

- Radius-Margin

- Rprop

- RBFKernel
  - #gamma

- SVM
  - #alpha
  - #b
Supervised learning with Shark

Application: Training a Support Vector Machine
Supervised learning with Shark

**Advantage**

We apply a single coherent optimization framework to different levels of inference!
Outline

- Introduction
- Supervised learning with Shark
- Roundtrip through all the rest
- Conclusions
Roundtrip through all the rest

There is a lot more around ...

- Many more Models, ErrorFunctions, Optimizers
There is a lot more around ...

- Many more Models,
  ErrorFunctions, Optimizers
- Evolutionary Algorithms (EAs)
There is a lot more around …

- Many more Models, ErrorFunctions, Optimizers
- Evolutionary Algorithms (EAs)
- Multi-Objective Optimization (MOO)
There is a lot more around ...

- Many more Models, ErrorFunctions, Optimizers
- Evolutionary Algorithms (EAs)
- Multi-Objective Optimization (MOO)
- Fuzzy-Logic
Shark on the web

- http://shark-project.sourceforge.net

The official Shark-Project site

Scope

SHARK is a modular C++ library for the design and optimization of adaptive systems. It provides methods for linear and nonlinear optimization, in particular evolutionary and gradient-based algorithms, kernel-based learning algorithms and neural networks, and various other machine learning techniques. SHARK serves as a toolbox to support real world applications as well as research in different domains of computational intelligence. The libraries are not necessarily dependent on any third party software. The sources are compatible with the following platforms: Windows, Solaris, MacOS X, and Linux.
Roundtrip through all the rest

Shark on the web

- http://shark-project.sourceforge.net
- http://mloss.org
Roundtrip through all the rest

Shark on the web

- http://shark-project.sourceforge.net
- http://mloss.org

Machine Learning Open Source Software

To support the open source software movement, JMLR MLOSS publishes contributions related to implementations of non-trivial machine learning algorithms, toolboxes or even languages for scientific computing. Submission instructions are available [here](http://mloss.org).

**A Library for Locally Weighted Projection Regression**
- [abs](http://jmlr.csail.mit.edu/jmlr/)
- [code](http://mloss.org)

**Shark**
- [abs](http://jmlr.csail.mit.edu/jmlr/)
- [code](http://mloss.org)

**LIBLINEAR: A Library for Large Linear Classification**
- [abs](http://jmlr.csail.mit.edu/jmlr/)
- [code](http://mloss.org)
Outline

- Introduction
- Supervised learning with Shark
- Roundtrip through all the rest
- Conclusions
Conclusions

What makes Shak different?

Three points to take home:
- Platform independent and self-contained
- Flexible modular design
- Covers multiple areas of machine learning
Conclusions

What makes Shak different?

Three points to take home:
Conclusions

What makes Shak different?

Three points to take home:

- platform independent and self-contained
What makes Shak different?

Three points to take home:

- platform independent and self-contained
- flexible modular design
Conclusions

What makes Shak different?

Three points to take home:

- platform independent and self-contained
- flexible modular design
- covers multiple areas of machine learning
Thank you for your attention!