Dr. Vishwanathan is senior researcher at NICTA's Canberra Laboratory, where he is a member of the Statistical Machine Learning research program.

The first laboratory on March 7 will feature some hands on experiments with Elefant (http://elefant.developer.nicta.com.au) mainly concentrating on installing, using, and developing machine learning algorithms within the Elefant framework. We will walk through examples of implementing a simple stochastic gradient descent algorithm as a part of this tutorial.

The second session on March 14 will feature hands on experiments with BNRM (Bundle Methods for Regularized Risk Minimization) (http://users.rsise.anu.edu.au/~chteo/BMRM.html). The emphasis here will be on developing various loss function modules which can then be plugged into the BMRM solver.

CD's and USB sticks containing installation instructions for Elefant and BMRM will be handed out during the session. Preferably bring your own laptops, although some spare PC's might be made available. Students will also have a chance to interact with the leading developers of Elefant and BMRM.
Welcome to Elefant

Visit us at:

Welcome to the Elefant home page. Elefant (Efficient Learning, Large-scale Inference, and Optimization Toolkit) is an open source library for machine learning licensed under the Mozilla Public License (MPL). We aim at developing an open source machine learning platform which will become the platform of choice for prototyping and deploying machine learning algorithms.

Elefant is developed by the SML group at National ICT Australia
What is Elefant

Elefant (Efficient Learning, Large-scale Inference, and Optimisation Toolkit) is an open source library for machine learning.

Elefant includes modules for many common optimisation problems arising in machine learning and inference. It is designed to be modular and easy to use. Framework provides easy to use python interface which can be use for quick prototyping and testing inference algorithms.

The diagram below illustrates the high-level system view of Elefant. For more information browse the documentation.

Applications

ELEFANT Toolkit in Python

Core ELEFANT Modules

Various machine learning algorithms
Support Vector Machines, Feature selections, Estimation, Classification, Quantile and Novelty detection, Neural Networks, Decision Trees, CRF, Graphical Models, etc.
Kernels - Linear, RBF, Dot Product, String, Multi class
Optimizers and Solvers
Algorithms written in C or C++ or JAVA or Fortran
Linear Algebra Library:
PyPETSc, PyTAO, PySLEPc, PyOOQP
Scipy and Numpy
Natural Language processing tools:
Interface to GateD and UIMA

External packages

Packages for extending C, C++, Fortran and JAVA code into python:
CTypesLib, SWIG, JPIPE, F2PY
C or C++ External Packages:
PETS, TAO, SLEPc, OOP
JAVA External Packages:
UIMA, GateD, Mailet
Fortran External Modules:
BLAS/LAPACK
What does Elefant Do?

Following are the key features of Elefant:

- Light weight component based system design, plug and play kind of a architecture
- Component suite for basic as well as advanced machine learning algorithms
- Support for various data source formats
- Components for data visualizations
- Easy to use graphical user interface for visual programming and quick prototyping
- Intuitive application programming interface for advanced prototyping
- Python wrappers for high-performance parallel scientific packages like PETSc, TAO, and SLEPc
- Interface to external systems like UIMA using jtype
- Comprehensive system documentation
- Open source and licensed under the Mozilla Public License (MPL)

Following machine learning algorithms are implemented in the Elefant:

- Support Vector Machine (SVM) for classification, regression, quantile and novelty detection, online learning, Epsilon Insensitive and Laplacian support vector regression.
- Gaussian Process Regression, Heteroscedastic Gaussian Process regression, Multi-class transductive classification with Gaussian Process.
- Solvers for the quadratic programming problem
- BAHISTIC feature selection
- Algorithms for fast computation and manipulation of kernel matrices. Linear, RBF, Dot Product and String Kernels
- Loopy Belief Propagation and Junction Tree algorithms.
- Cover Tree for calculating the nearest neighbor

For future plans browse the Roadmap.
Notice

A new version of BMRM with more complete features is coming soon. This version is mainly for experimental use.

Overview

BMRM is an open source, modular and scalable convex solver for many machine learning problems cast in the form of regularized risk minimization problem [1]. It is "modular" because the (problem-specific) loss function module is decoupled from the (regularization-specific) optimization module (e.g. quadratic programming or linear programming solvers), thus shorten the time to implement/prototype solutions to new problems. Besides, the decoupling leads to easier parallelization of the loss function computation. At the moment, BMRM can solve the following problems (and more problems soon),

1. Binary classification
   - Soft-margin [2,7]
   - Squared Soft-margin [3]
   - Huber-hinge [3]
   - Logistic regression [4]
   - Exponential [5]
   - ROC Score [6]
   - $F_{\beta}$ Score [6]

2. Univariate regression
   - $\epsilon$-insensitive [8]
   - Huber robust [9]
   - Least Mean Squares [10]
   - Least Absolute Deviation

3. Novelty detection (1-class SVM) [11]
4. Quantile regression [12]
5. Poisson regression [13]
6. Ranking
   - NDCG (normalized discounted cumulative gain) [14]
   - Ordinal regression [15,6,7]

along with either $l_1$ or $l_2$ regularizer. Also, users can add new loss function for problems with structured input and output variables.

Download

BMRM version 1.0
Installation

BMRM version 1.0 requires the following external (publicly available) libraries:

- **PETSc** for matrix operations.
  (Suggested configuration flags: --download-c-blas-lapack=ifneeded --with-clanguage=C++ --CXXFLAGS=-O3 --with-debugging=no)
- **MPICH2** for parallel computation.
- **COIN-OR Clp** as linear programming solver

Prior to the installation of BMRM, the above libraries must be installed properly. An example of installation of BMRM (including the required external libraries) on a linux machine is presented in the README file in the source code bundle.

Disclaimer

BMRM is licensed under [Mozilla Public License version 1.1](https://www.mozilla.org/en-US/MPL/1.1/). The authors are not responsible for any implications from the use of the software.

FAQs, Suggestions, Comments, Bug Reports

(You are welcomed to expand this list ;-) )

Contacts

[Choon Hui Teo](mailto:teo@cs.nyu.edu) | [Quoc Le](mailto:quocle@cs.columbia.edu) | [Alex Smola](mailto:alex.smola@MPI-INF.DE) | [SVN Vishwanathan](mailto:vishv@cs.umd.edu) | [Markus Weimer](mailto:mmweimer@cs.nyu.edu)

References


Last modified: 10 August 2007