GURLS:
A Least Squares Library for Supervised Learning

Alessandro Rudi, Lorenzo Rosasco

(Matteo Santoro, Andrea Schiappacasse, Carlo Ciliberto)

Humanoids 2014, IEEE-RAS International Conference on Humanoid Robots

November 18th 2014 – Madrid

Summary

- Machine Learning
- GURLS: Basic Facts and Ideas
- Design and Architecture of the Library
- Selected Experiments and Results

Machine Learning in a Nutshel

A share of big data
(+ big computers)

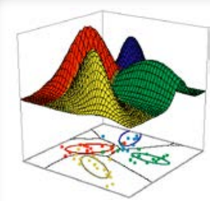
+

"Favorite"
preprocessing

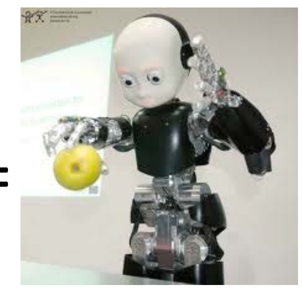
Data
representation

+

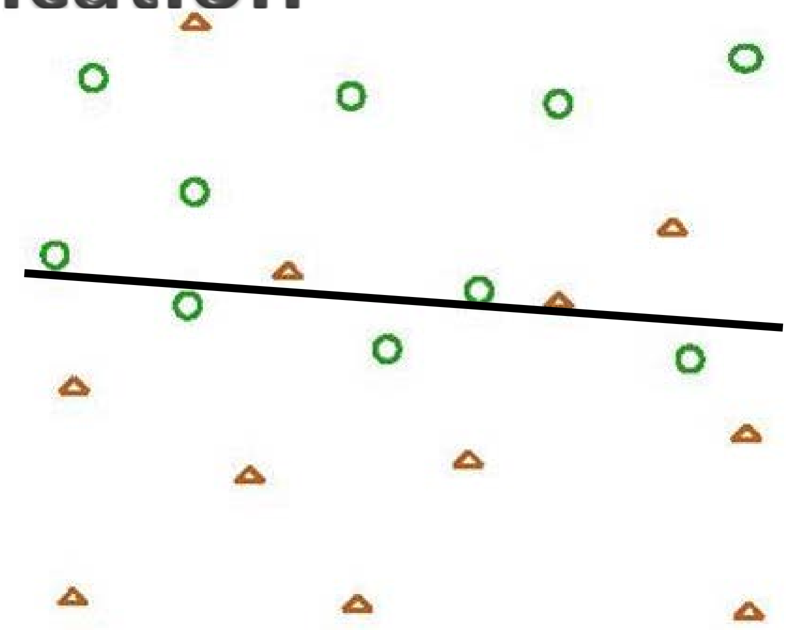
Statistical
modeling



=



Basic Setting: Classification



Find

$$f(x) \sim y$$

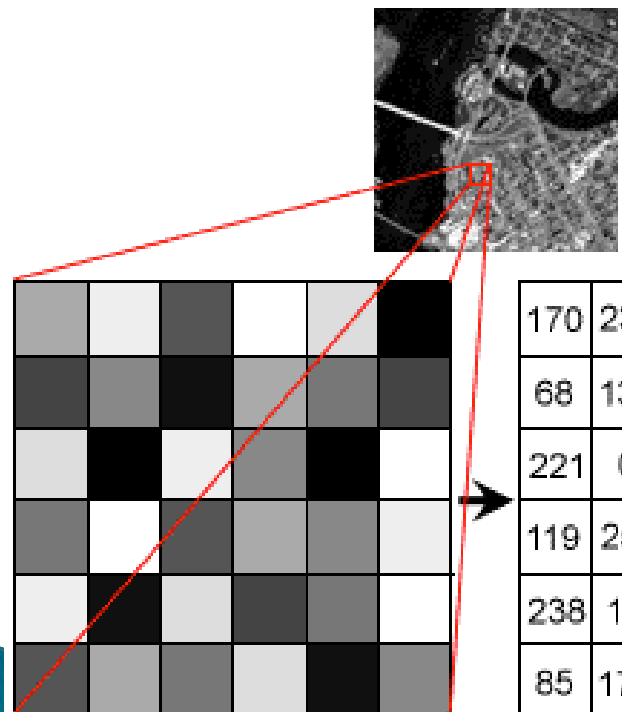
$$(x_1, y_1), \ldots, (x_n, y_n)$$

$$x_i \in \mathbb{R}^p \text{ and } y_i \in Y = \{-1, 1\}, i = 1, \ldots, n$$

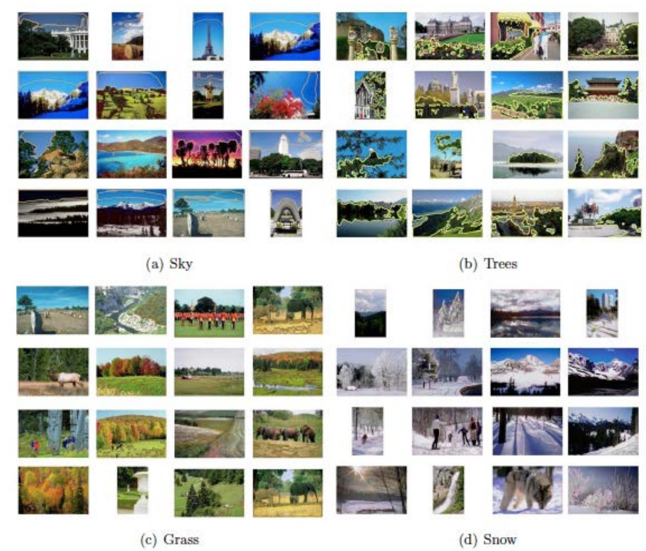
$$X_n = \begin{pmatrix} x_1^1 & \ldots & \ldots & \ldots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \ldots & \ldots & \ldots & x_n^p \end{pmatrix} \qquad Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Image Classification

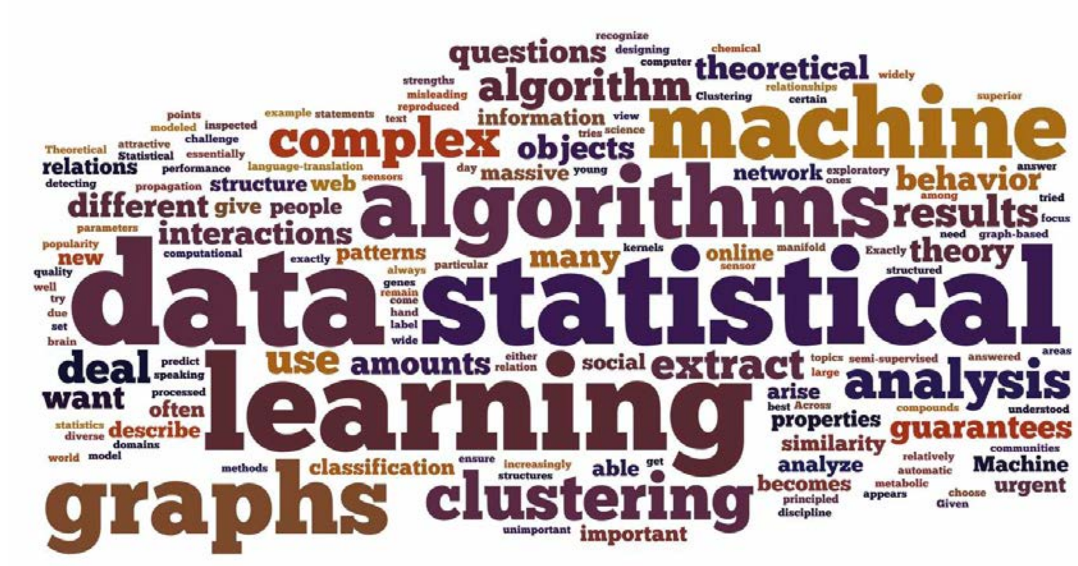
$$X_n = \begin{pmatrix} \boxed{x_1^1 \quad \dots \quad \dots \quad \dots \quad x_1^p} \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \\ x_n^1 \quad \dots \quad \dots \quad \dots \quad x_n^p \end{pmatrix}$$

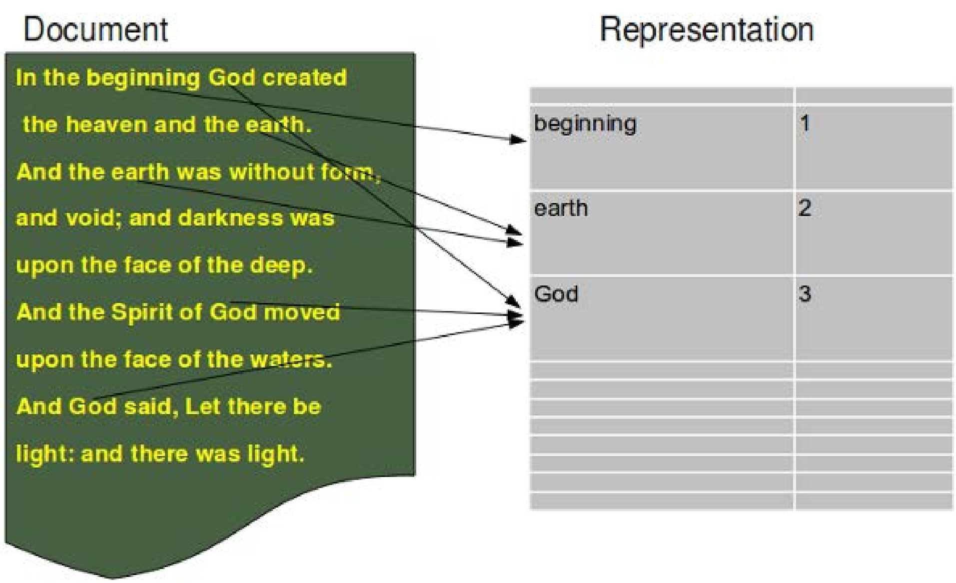


170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136



Text Classification: Bag of Words





$$X_n = \begin{pmatrix} \boxed{x_1^1 \quad \dots \quad \dots \quad \dots \quad x_1^p} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ x_n^1 \quad \dots \quad \dots \quad \dots \quad x_n^p \end{pmatrix}$$

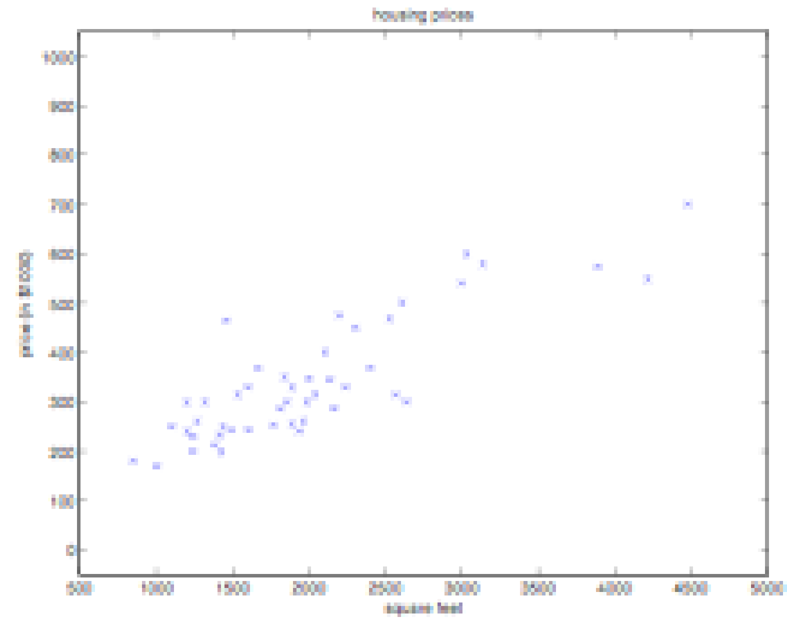
From classification to regression

$$(x_1, y_1), \ldots, (x_n, y_n)$$

$x_i \in \mathbb{R}^D$ and $\cancel{y_i \in Y = \{-1, 1\}, i = 1, \ldots, n}$

$$y_i \in Y \in \mathbb{R}, i = 1, \ldots, n$$

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



Living area (feet^2)	#bedrooms	Price (1000\$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

$$y_i = f(x_i) + \sigma\varepsilon_i, \quad \sigma > 0$$

e.g. $f(x) = w^T x, \quad \varepsilon_i \sim N(0, 1)$

Variations on a Theme

$$(x_1, y_1), \ldots, (x_n, y_n)$$

Multiclass: $x_i \in \mathbb{R}^D$ and $y_i \in Y = \{1, \ldots, T\}, i = 1, \ldots, n$

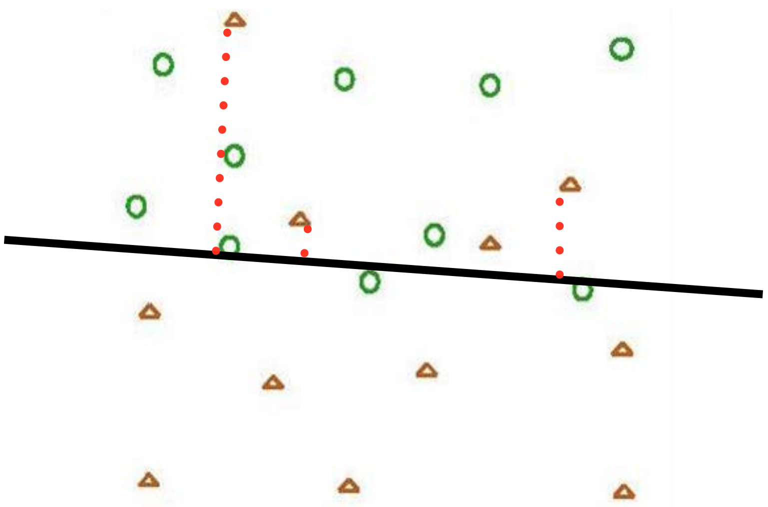
Multitask: $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}^T, i = 1, \ldots, n$

Empirical error

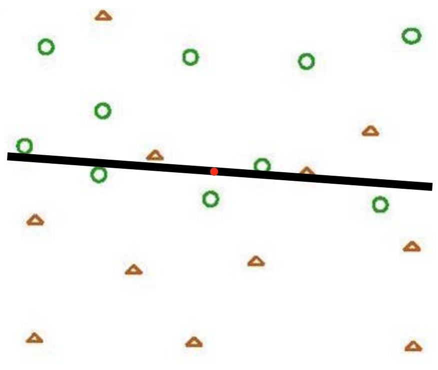
Given a loss function $V(f(x), y)$

We can define the Empirical Error

$$I_S[f] = \frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$



Hypotheses Space



parametric

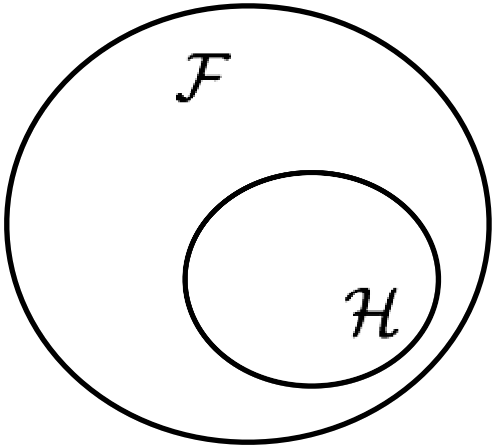
- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$
- **Generalized linear models** $f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j$

non-parametric

- **Reproducing kernel Hilbert spaces** $f(x) = \sum_{j\geq 1} \Phi(x)^j w^j = \sum_{i\geq 1} K(x, x_i)\alpha_i$

$K(x, x')$ is a symmetric positive definite function called reproducing kernel

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$



Kernels

- **Linear kernel**

$$K(x, x') = x^T x'$$

- **Gaussian kernel**

$$K(x, x') = \mathrm{e}^{-\frac{\|x - x'\|^2}{\sigma^2}}, \qquad \sigma > 0$$

- **Polynomial kernel**

$$K(x, x') = (x^T x' + 1)^d, \qquad d \in \mathbb{N}$$

- **Inner Product kernel/Features**

$$K(x, x') = \sum_{j=1}^{p} \Phi(x)^j \Phi(x')^j \qquad \Phi : X \to \mathbb{R}^p.$$

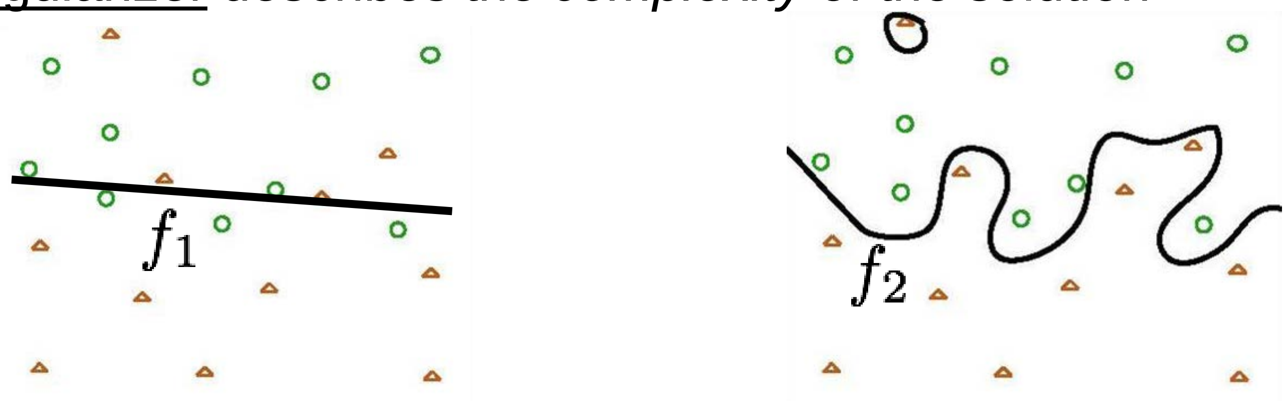
(Tikhonov) Regularization

regularization parameter

$$\min_{f\in\mathcal{H}}\{\frac{1}{n}\sum_{i=1}^{n}V(y_i,f(x_i))+\lambda R(f))\}\to f_S^{\lambda}$$

regularizer

- The <u>regularizer</u> describes the *complexity* of the solution



$R(f_2)$ is bigger than $R(f_1)$

- The <u>regularization parameter</u> determines the trade-off between complexity and empirical risk

(Tikhonov) Regularization & Stability

Consider $f(x) = w^T x = \sum_{j=1}^{p} w^j x^j$, and $R(f) = w^T w$,

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \longrightarrow w_n = (X_n^T X_n)^{-1} X_n^T Y_n$$

$$\downarrow \qquad\qquad \downarrow$$

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2 \right\} \qquad w_n^\lambda = (X_n^T X_n + \lambda n I)^{-1} X_n^T Y_n$$

Online & Incremental Learning

f_0

$f_1 \leftarrow (x_1, y_1)$

$f_2 \leftarrow (x_2, y_2)$

$\dots$

$f_n \leftarrow (x_n, y_n)$

$\dots$

$(x_1, y_1), \dots, (x_n, y_n)$

f_0

$f_1 \leftarrow (x_1, y_1)$

$f_2 \leftarrow (x_2, y_2)$

$\dots$

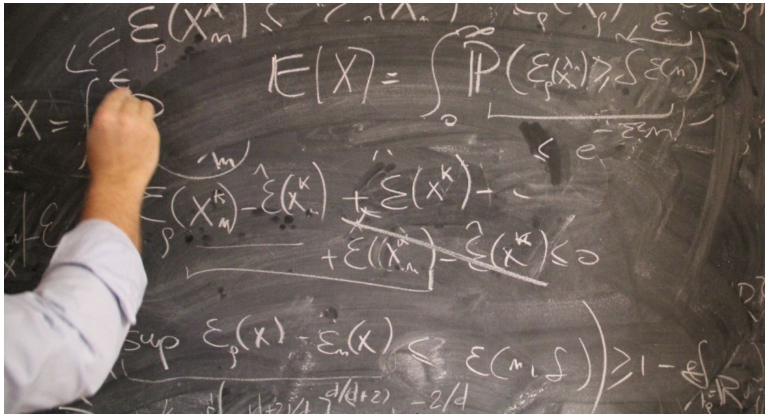
$f_n \leftarrow (x_n, y_n)$

$f_1 \leftarrow (x_1, y_1)$

$\dots$

The Library

GURLS: a Least Squares Library for Supervised Learning.
Andrea Tacchetti, Pavan K Mallapragada, Matteo Santoro, Lorenzo Rosasco.
The Journal of Machine Learning Research. Volume 14, 3201-3205, 2013.



- http://lcsl.mit.edu
- https://github.com/LCSL/GURLS

The Library

- **GURLS**
  - a MATLAB toolbox for regression and (multiclass) classification based on the Regularized Least Squares (RLS) loss function. Datasets that fit into your computer's memory should be handled with this package.
- **bGURLS** (b is for big)
  - a MATLAB toolbox to use RLS on very large matrices by means of memory-mapped storage and a simple distributed task manager.
- **GURLS++**
  - a C++ standalone implementation of GURLS, with additional simple API's for specific learning pipelines
- **bGURLS++**
  - a C++ standalone implementation of bGURLS.

Key Requirements

- **Speed**
  - Fast training/testing procedures (online, batch, randomized, distributed) for learning problems with potentially very large number of points, features and outputs (classes/variables).
- **Memory**
  - Flexible data management to work with large datasets by means of memory-mapped storage.
- **Performance**
  - State of the art results in high-dimensional multi-output problems (e.g. object recognition tasks with tens or hundreds of classes, where the input have dense features).
- **Usability and modularity**
  - Easy to use and to expand library.

Available Tasks in GURLS

Task category	Description	Available tasks
split	Splits data into one or more pair of training and validation sets	ho
paramsel	Performs selection of the regularization parameter lambda and, if using Gaussian kernel, also of the kernel parameter sigma	fixlambda, loocvprimal, loocvdual, hoprimal, hodual, siglam, siglamho, bfprimal, bfdual, calibratesgd, hoprimalr, hodualr, horandfeats, gpregrLambdaGrid, gpregrSigLambGrid, loogpregr, hogpregr, siglamhogpregr, siglamloogpregr,
kernel	Builds the symmetric kernel matrix to be used for training	chisquared, linear, load, randfeats, rbf
rls	Computes the optimizer	primal, dual, auto, pegasos, primalr, dualr, randfeats, gpregr
predkernel	Builds the train-test kernel matrix	traintest
pred	Predicts the labels	primal, dual, randfeats, gpregr
perf	Assess prediction performance	macroavg, precrec, rmse
conf	Computes a confidence for the highest scoring class	maxscore, gap, boltzmangap, boltzman

The «pipeline» concept

- A sequence of «tasks»:
  - *{'kernel:linear', 'paramsel:loocvdual', 'rls:dual', 'pred:dual', 'perf:macroavg'}*
- A sequence of processing policies:
  - *{ C1, C2, C3, C4, C5}*
- A processing policy “Ci” may be:
  - 0 = Ignore
  - 1 = Compute
  - 2 = Compute and save
  - 3 = Load from file
  - 4 = Explicitly delete

The «pipeline» concept

- GURLS includes an «engine» responsible for the parsing of the pipeline description and execute a whole machine learning experiment.

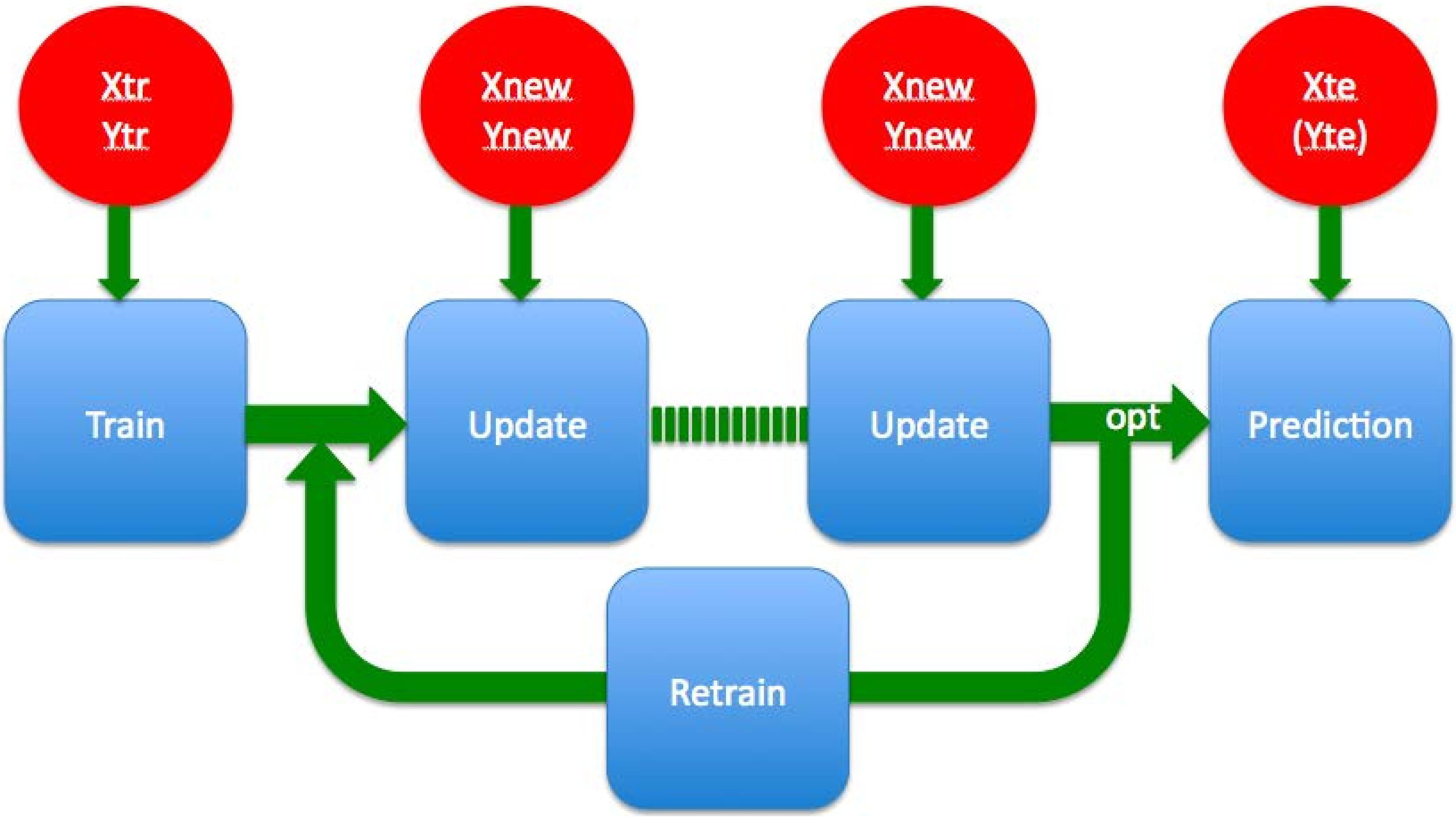
```
name = 'ExampleExperiment';
opt = defopt(name);
opt.seq = {'paramsel:loocvprimal','rls:primal','pred:primal', ...
           'perf:precrec','perf:macroavg'};
opt.process{1} = [2,2,0,0,0];

opt.process{2} = [3,3,2,2,2];

gurls (Xtr, ytr, opt,1);

gurls (Xte, yte, opt,2);
```

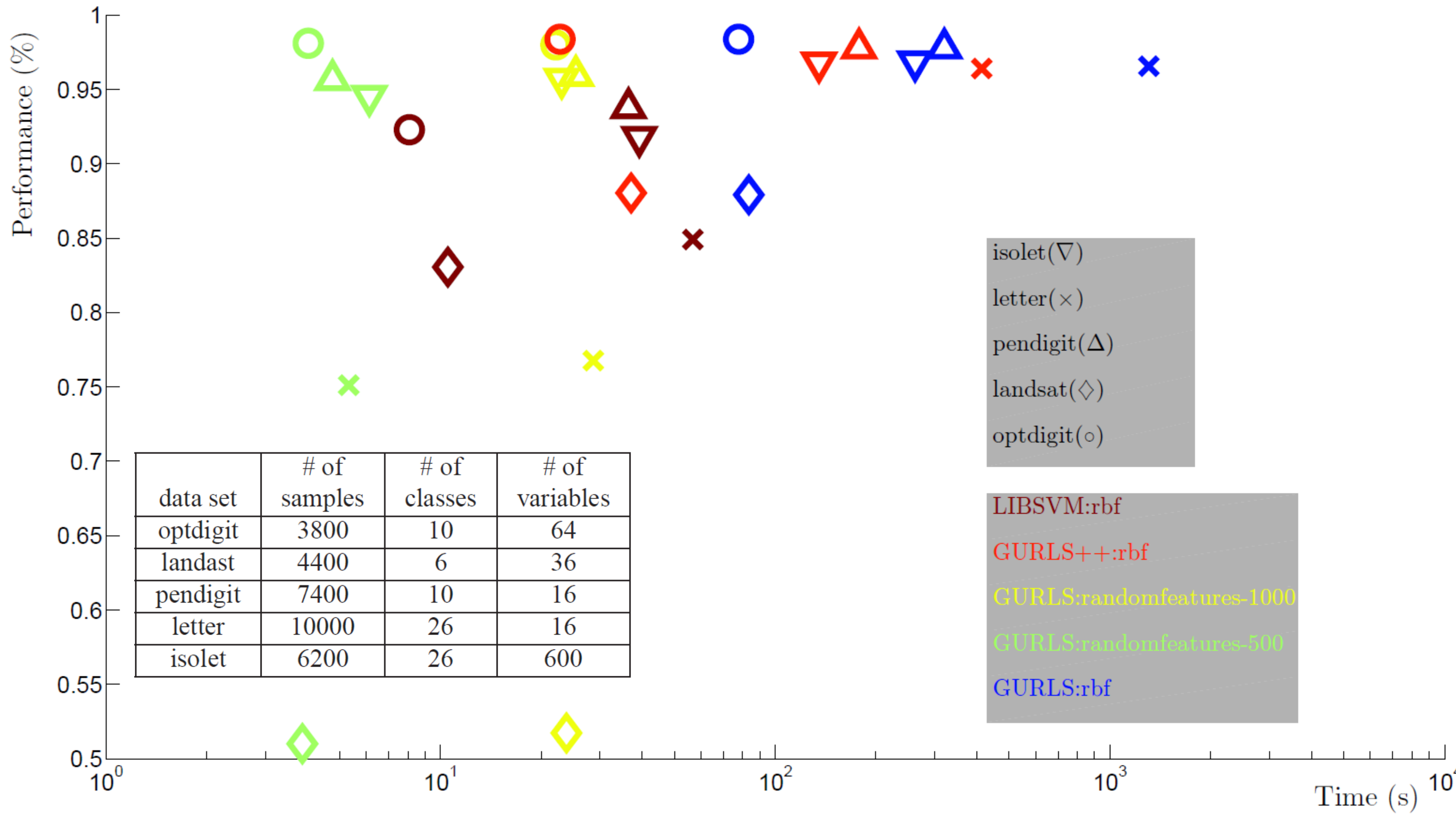
Beyond the pipeline…



GURLS++

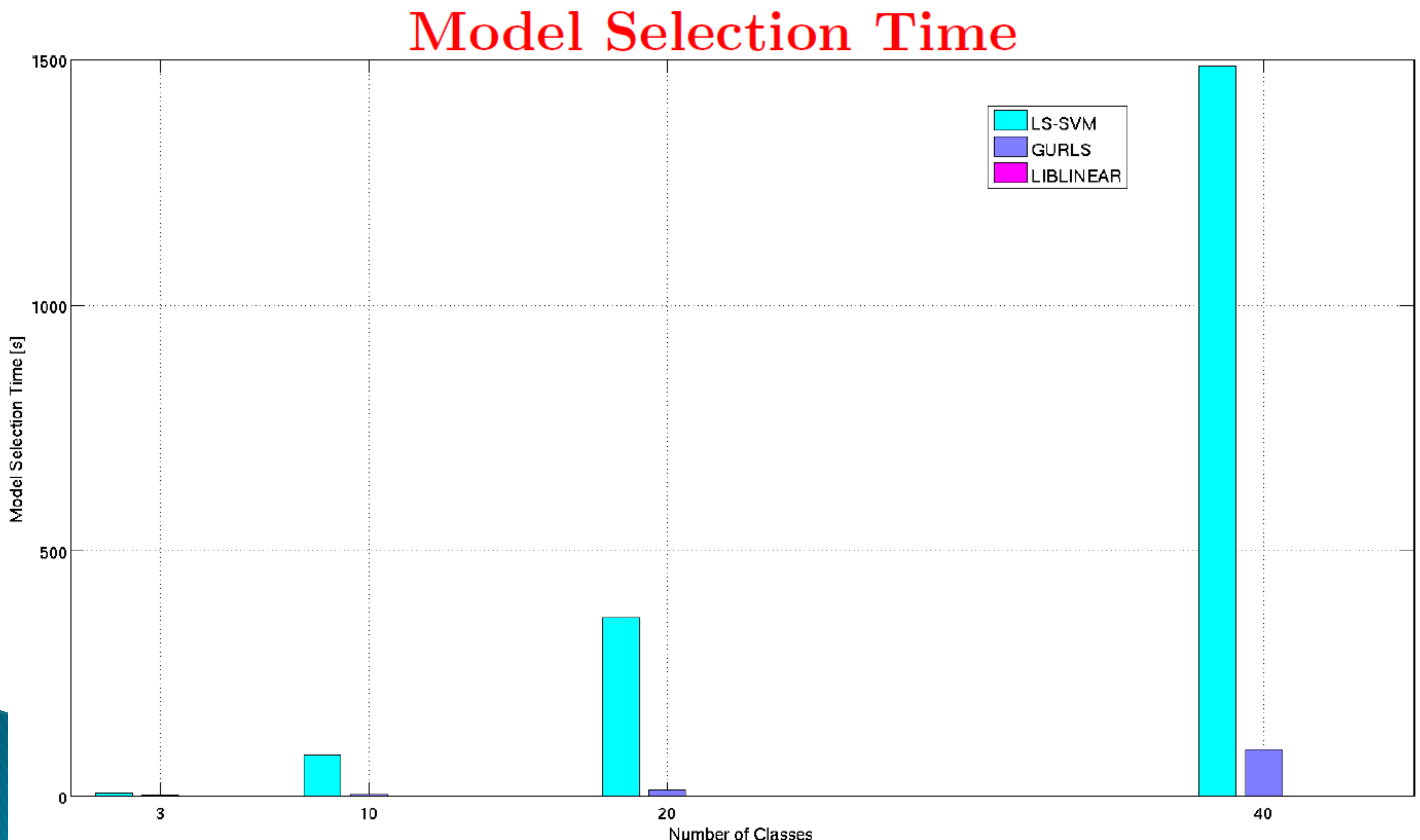
- A complete (and revised!) C++ porting of the original Matlab package.
- (Only) a bit more complicated to install. Superbuild (Windows and Unix) available for installation.
- Requires some external dependecies:
  - Boost (mainly for serialization)
  - Blas/Lapack (you may choose the one you prefer)

Experimental Results

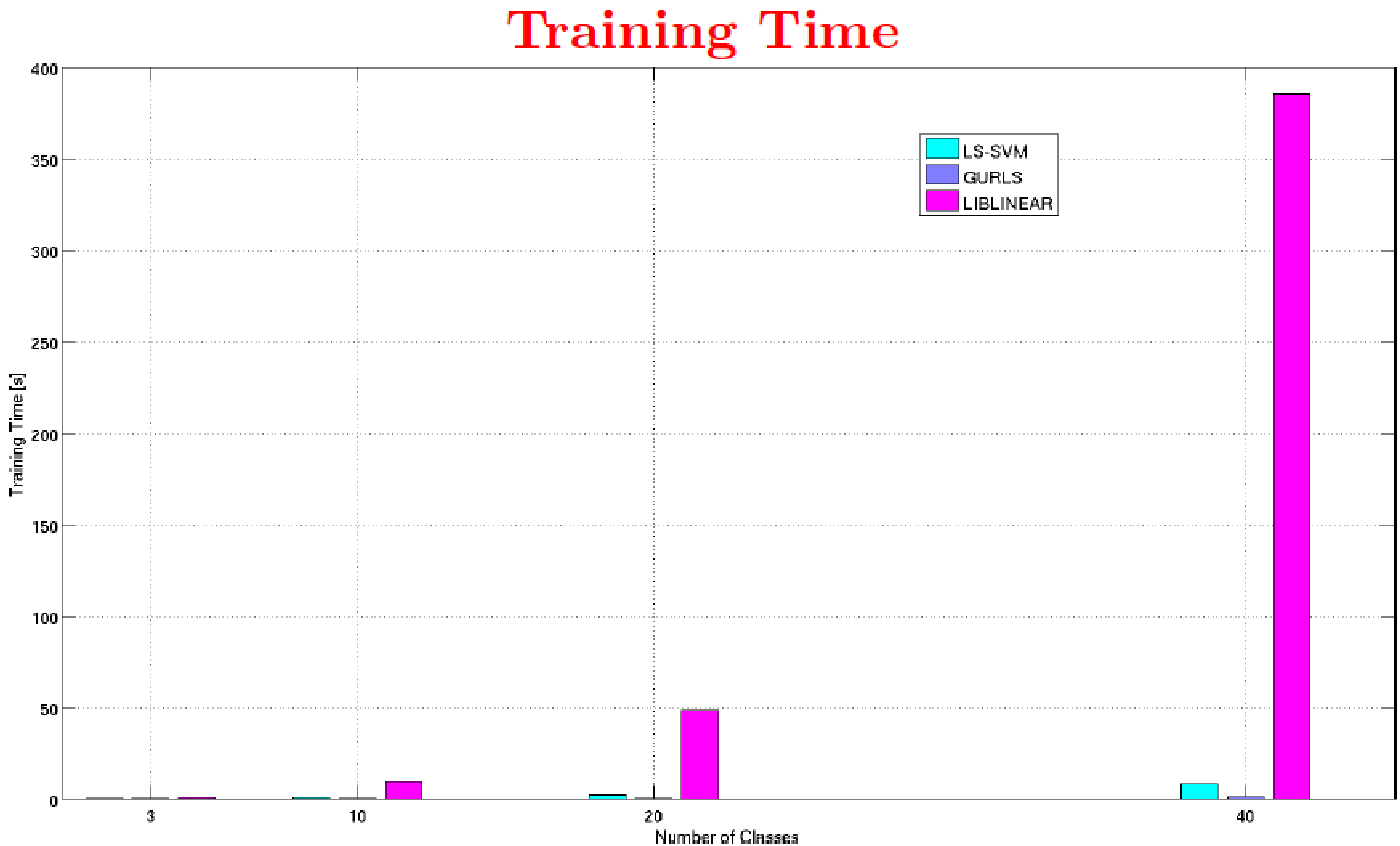


data set	# of samples	# of classes	# of variables
optdigit	3800	10	64
landast	4400	6	36
pendigit	7400	10	16
letter	10000	26	16
isolet	6200	26	600

Experimental Results



Experimental Results



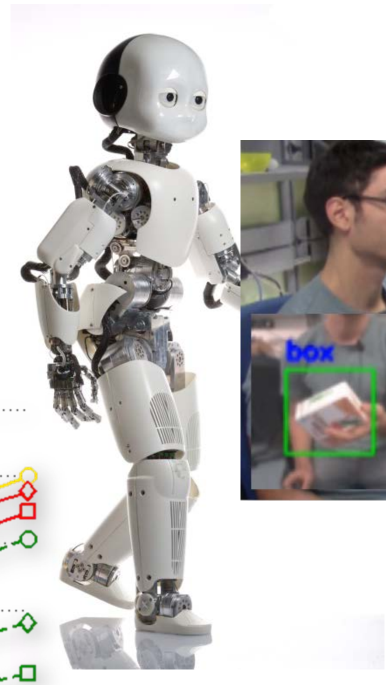
Experimental Results

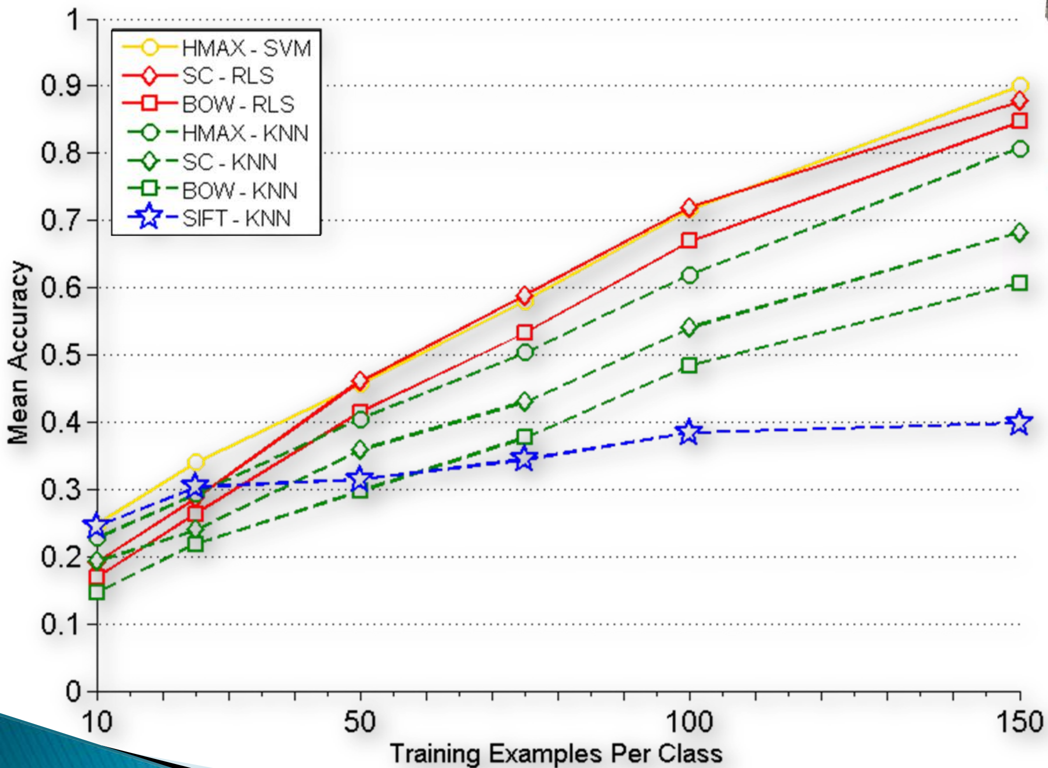
PubFig83

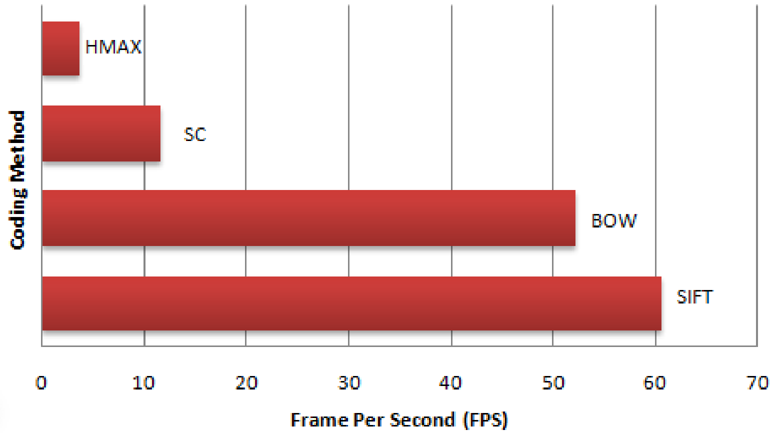
http://www.eecs.harvard.edu/ zak/pubfig83/

Package	*Kernel*	*Accuracy*	*Time*
GURLS	Linear	87%	0h13m
LIBSVM	Linear	76%	5h20m
GURLS	RBF with selection	88%	5h51m
GURLS	RBF = 25th PCT of DST	87%	0h14m
LIBSVM	RBF = 25th PCT of DST	76%	4h18m

A robotic example







C Ciliberto, S Fanello, M Santoro, L Natale, G Metta and L Rosasco. On the Impact of Learning Hierarchical Representations for Visual Recognition in Robotics. To appear in the Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)

Contacts

- alessandro.rudi@iit.it
- matteo.santoro@camelotbio.com
- andrea.schiappacasse@camelotbio.com

thank you!