Experiment Report

Name: RRLS_bench_icubdyn_RF_nMSE
Number: 5
Date: 2014-07-17

Author: Raffaello Camoriano (raffaello.camoriano@iit.it)

Goal

Within the GURLS framework for MATLAB, evaluate the following performances:

- Accuracy (nMSE) of the GURLS native batch estimator with RF mapping of the input space
- Accuracy (nMSE) of the recursive Cholesky estimator with RF mapping of the input space and initial lambda tuning, fixed sigma and fixed feature scaling.
- Accuracy (nMSE) of Arjan's batch and recursive Cholesky estimator with RF mapping of the input space and initial lambda and feature scaling factors tuning.

Data

This experiment uses the iCubDyn dataset, projected over a *numRF*-dimensional random features space. The RF projection used here is the standard GURLS one:

```
W = sqrt(2)*randn(d,numRF);
G = [cos(W*X);
     sin(W*X)];
```

Note: In order to save memory, the following projection scheme could be used:

$$\phi \leftarrow \sqrt{\frac{2}{D}} \cos\left(\boldsymbol{\Omega x} + \boldsymbol{\beta}\right)$$

```
W = randn(numRF,d);
b = 2*pi*rand(D,1);
V = proj.W*X;
for i = 1:D
   V(:,i) = V(:,i) + proj.b;
end
G = sqrt(2/D)*cos(V);
```

A comparative experiment will be set up to evaluate differences in time and memory complexity and in accuracy.

The samples are not randomized, since the incremental solution is mathematically proven to match the batch one and the RF approximation is assumed to be sufficiently accurate.

The data is split into training and test set. Each set is in the form of an input data matrix and an output labels vector.

p = 15000: Number of training samples, for parameter selection and initial training
n_te = 72850: Number of test samples

Algorithms

Batch

The batch algorithm is the standard one, documented in the GURLS manual.

- Hyperparameter λ is selected among 200 guesses via hold-out. Validation size: 20% of the training set.
- Training is then performed on the first p samples.
- Testing is performed as follows: For each test sample, a prediction is performed. Then, at each step *i*, the nMSE for each output is computed and saved according to:
  - $nMSE = 1 - R^2 = \frac{SS_{res}}{SS_{tot}} = \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} = \frac{\sum(y_i - \hat{y}_i)^2}{\sigma^2}$
  - R^2 is the coefficient of determination of the learned model
  - $nSE_0 = 0;$
  - $nSE_i = nSE_{i-1} + \frac{(y_i - \hat{y}_i)^2}{\sum_{k=1}^{n_{te}}(y_k - \bar{y})^2} = nSE_{i-1} + \frac{(y_i - \hat{y}_i)^2}{\sigma^2}$
  - $nMSE_i = \frac{nSE_i}{i}$

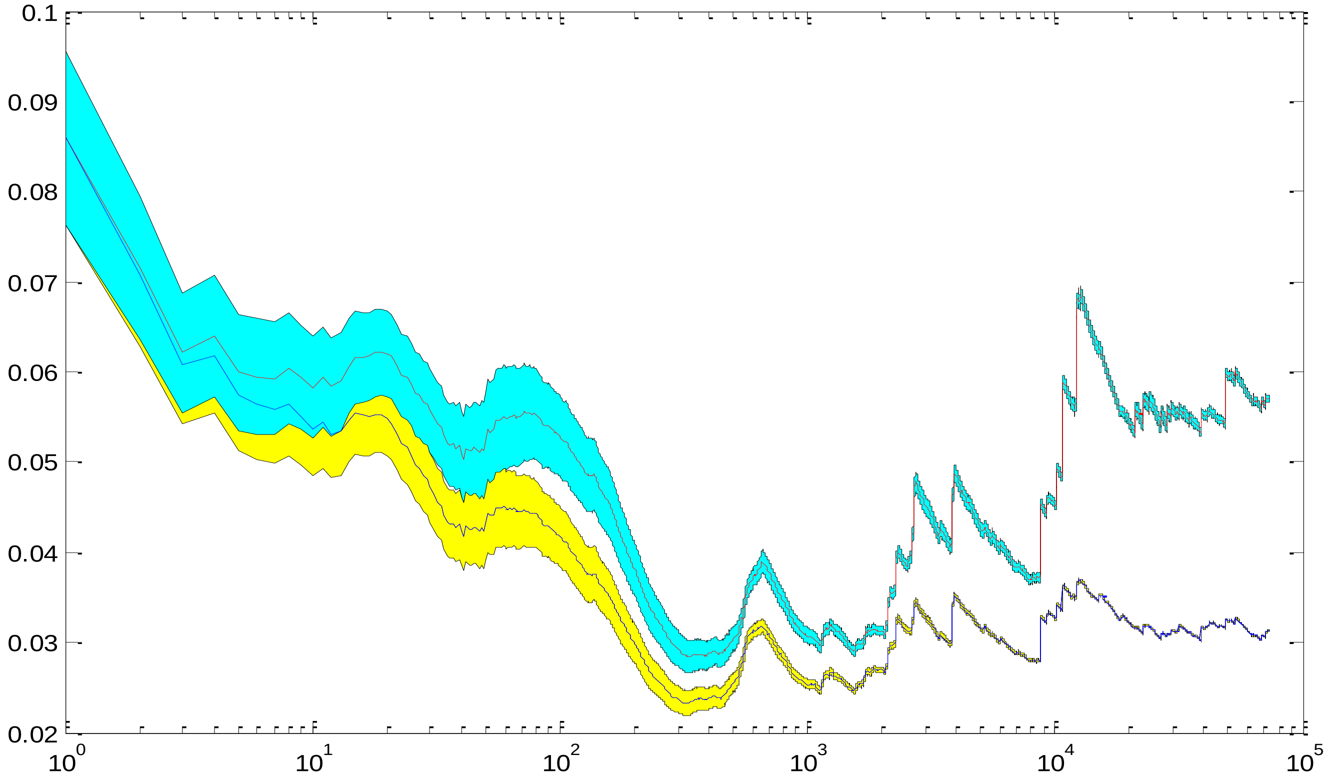
Recursive RLS

- Hyperparameter λ is the same one selected by the batch algorithm.
- The estimator is initialized over the first *p* samples, according to:
  ```
  1. R = chol(XtX + (n*lambda)*eye(d));
  2. b = Xty;
  3. W = R\(R'\Xty);
  ```
- Test-update phase
  - The estimator produces a prediction $\hat{y}_i$ for the i-th test sample.
  - $nMSE_i$ is computed and stored
  - The Cholesky factor *R*, *b* and *W* are updated with the new sample, according to:
    ```
    1. b = b + X(i,:)'*y(i,:);
    2. R = cholupdate(R,X(i,:)');
    3. W = R\(R'\b);
    ```

Results

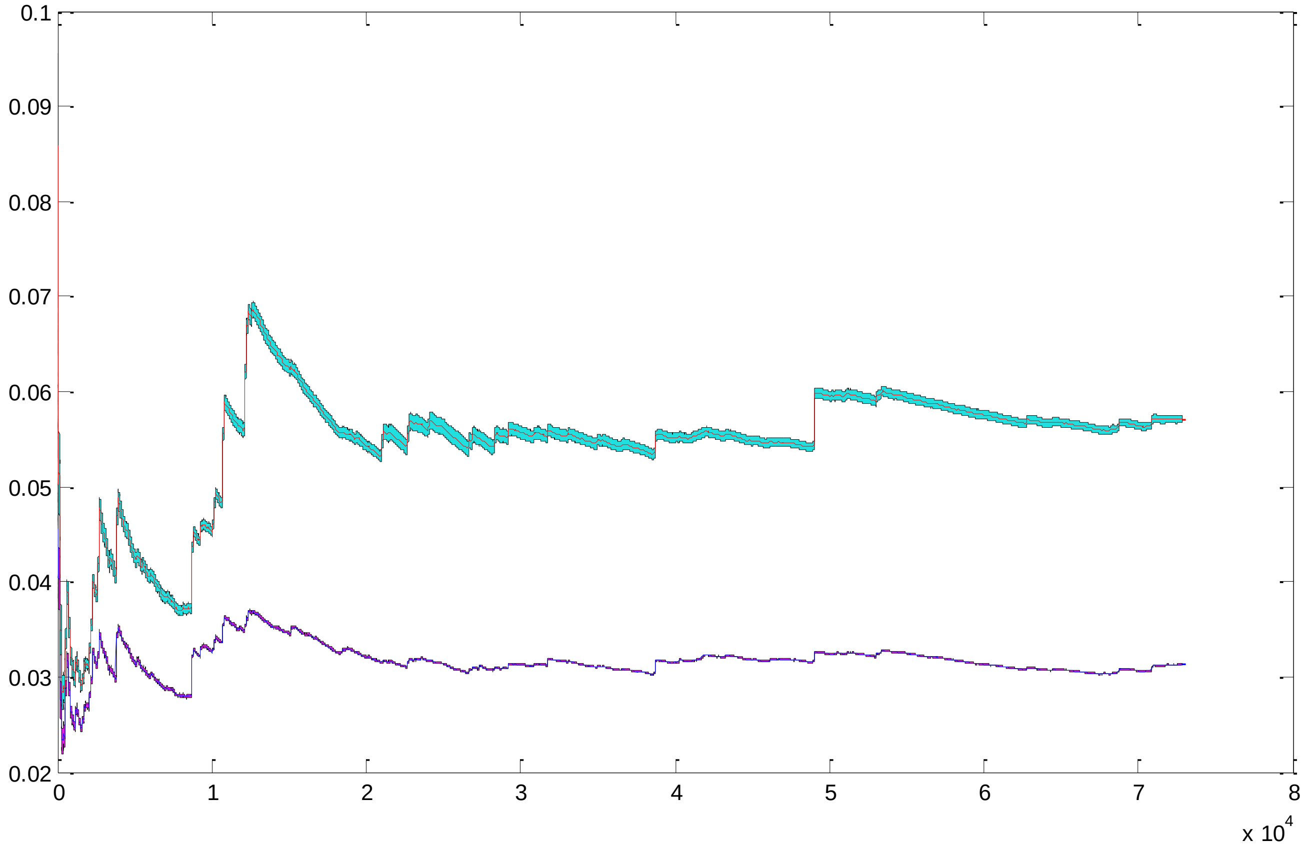
1) nMSE over the test set for numRF=500

Log scale



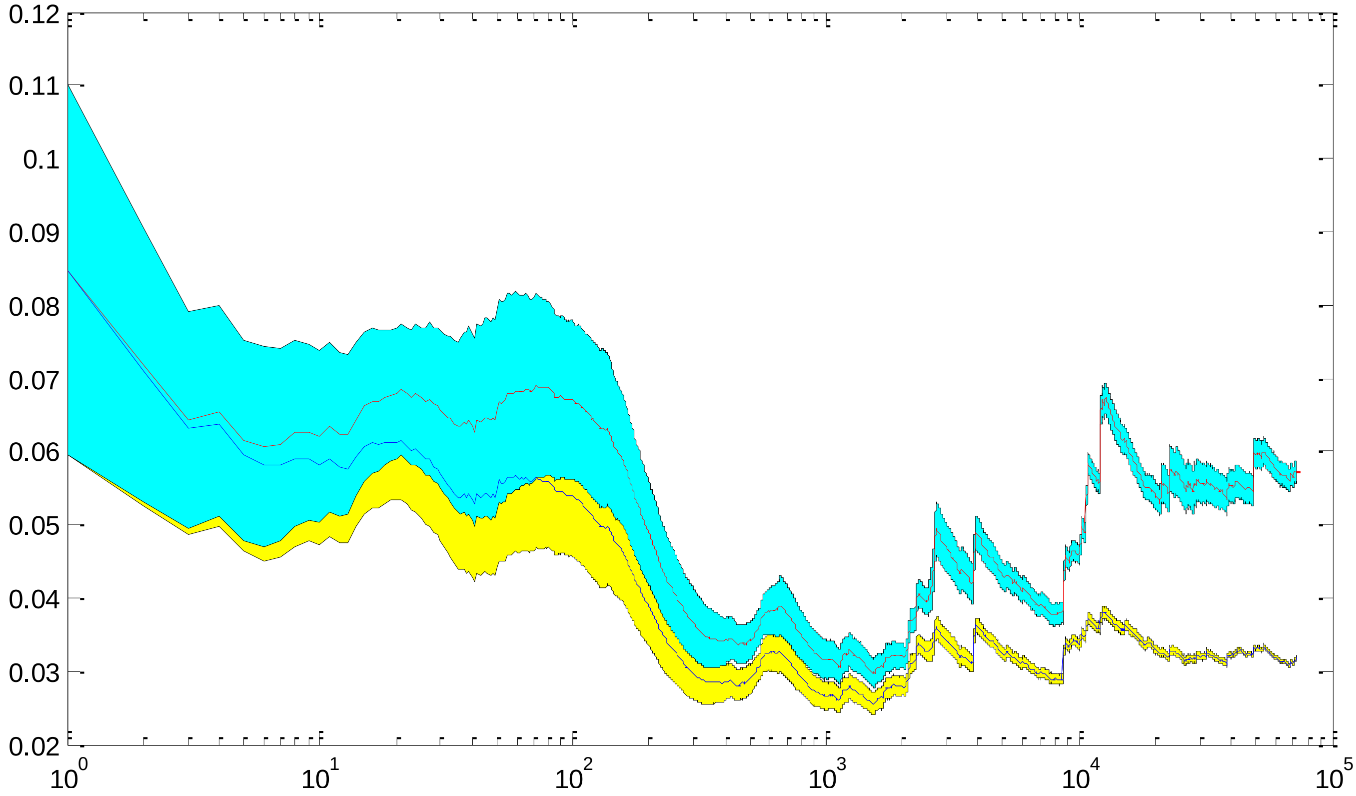
Red: batch; Blue: Recursive

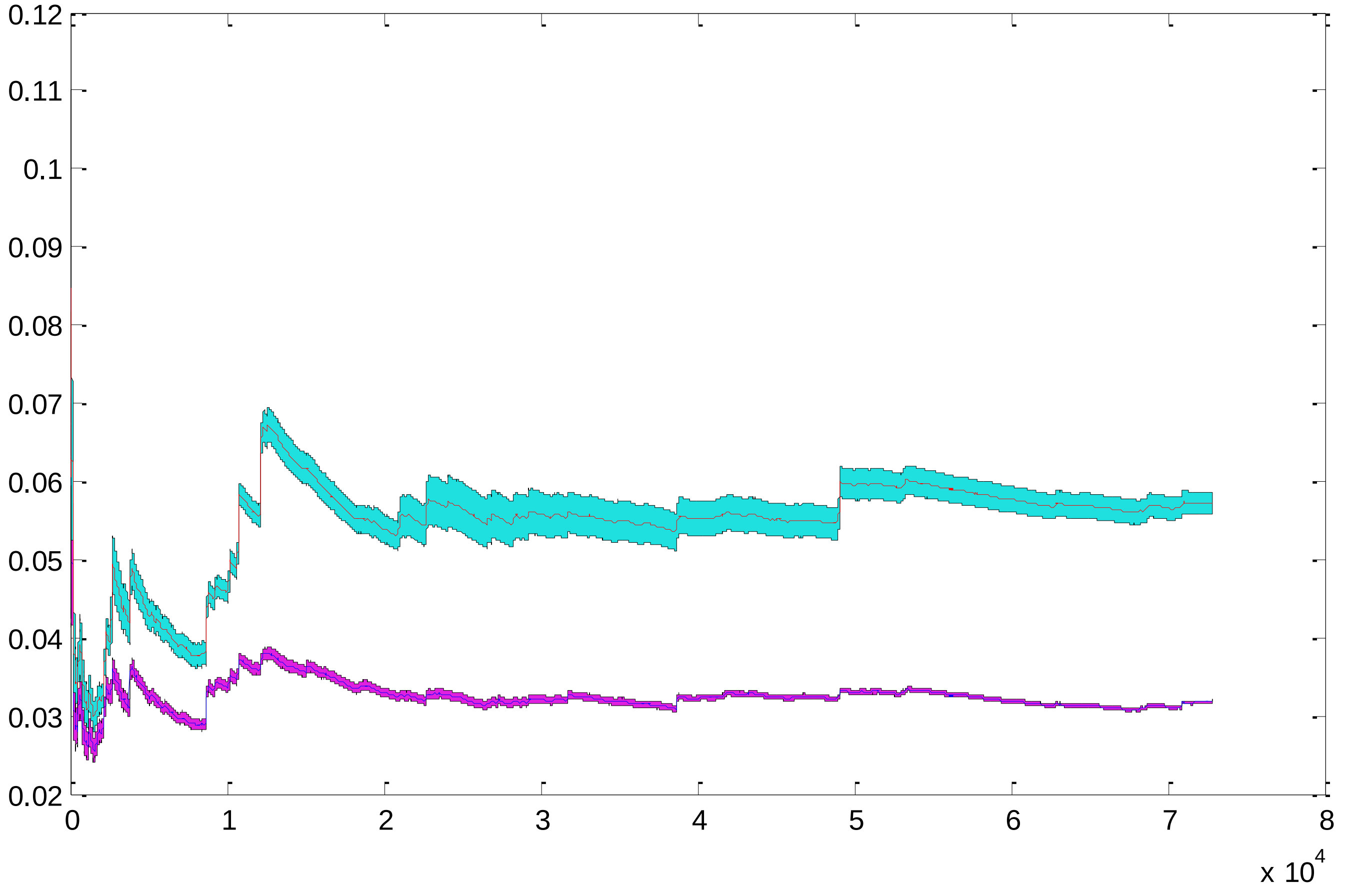
Linear scale



Red: batch; Blue: Recursive

2) nMSE over the test set for numRF=200





Discussion

Accuracy

Accuracy has been measured using the average nMSE formula reported above. nMSE is normalized by the output variance, computed in advance over the whole dataset.

Batch

nMSEs for the batch case essentially have the same behavior, and stabilize between 0.05 and 0.06. This happens even if the σ parameter(s) is (are) not optimized in the experiment employing GURLS. This may indicate that the choice of σ could be of secondary importance, but further investigation is needed to verify this statement.

Recursive

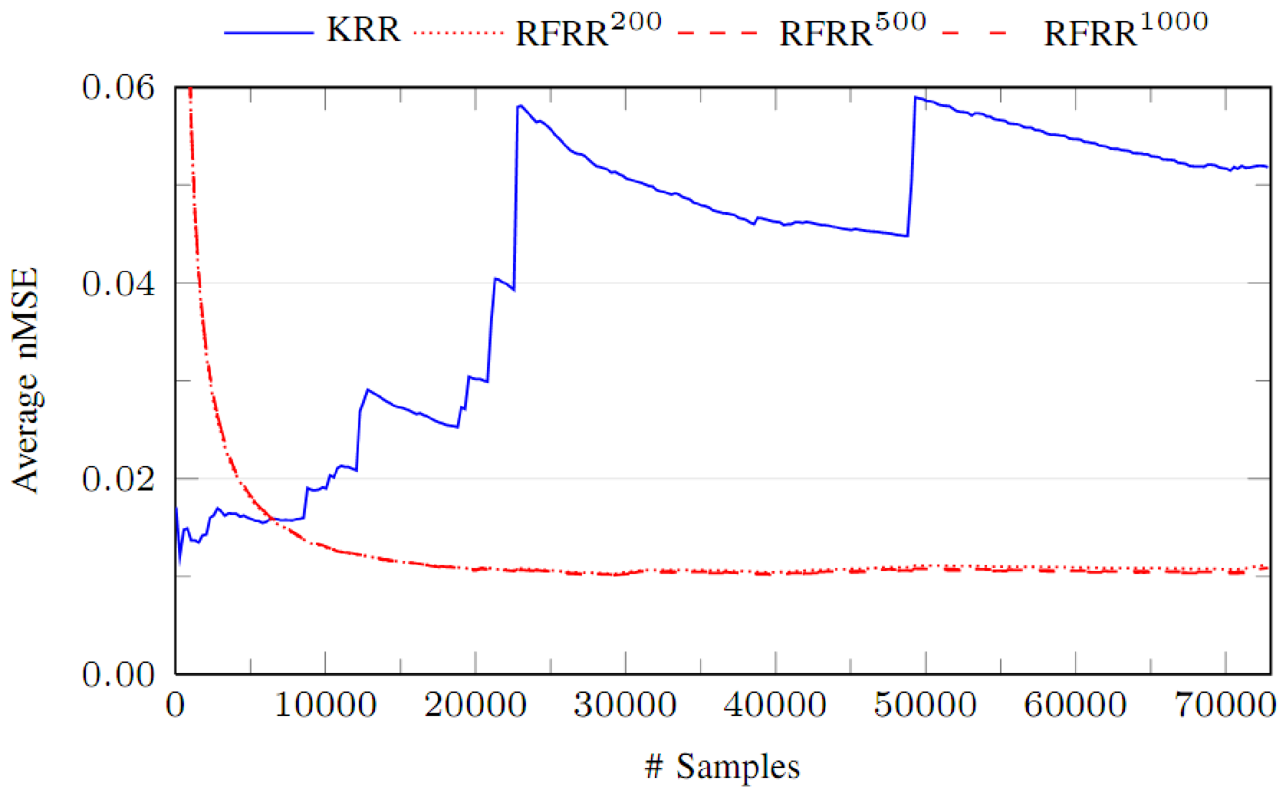
In the recursive case, Arjan does not initialize the recursive estimator with the training set used for the batch estimator training. On the other hand, in the GURLS experiment the estimator is initialized. The nMSE after the transient stabilizes to 0.01 in Learning Machine, and to 0.03 in GURLS. The discrepancy could be a result of the lack of sigma optimization.

Regularization

The optimal regularization parameter λ is in the order of e-10 (e.g. 2.48e-10) for numRF = 200 and numRF = 500, considering 200 guesses in the standard range specified by GURLS. This lambda value corresponds to the smallest one in the guesses vector, indicating that the regularization effect of the random features mapping renders Tikhonov regularization of no use in this case.
On the other hand, it is worth noting that GURLS only optimizes the lambda parameter, while in the Learning Machine experiments a marginal likelihood optimization of the scaling factor for each feature was carried out, corresponding (?) to the fine-tuning of the σ parameter(s) of the Gaussian kernel. This is a possible reason for the fact that the *nMSE* results reported by Arjan outperform the ones obtained using the GURLS framework. However, this discrepancy is rather small.

- Shall we tune a single σ for the RF approximation of the Gaussian kernel in GURLS?
- One for each feature?
- None?

Arjan's results



Conclusions and future work

- The effect of lambda on the results is negligible, due to the regularization effect of the random features mapping.
- The role of sigma shall be further investigated.
- Scaling the experiment up for using 1000 random features or more requires more computational power and memory than the one employed for this experiment. Anyway, it seems unlikely that accuracy will increase significantly, since scaling up from 200 to 500 RF did not result in a significant improvement.
- The average nMSE discrepancy at regime is rather limited (0.03 vs 0.01), suggesting that a prototype can be developed on the real robot.
- A test with the classical RF mapping (*W, b*) shall be made.
- A further experiment shall be performed there the GURLS recursive estimator is be initialized from scratch like the LM one and then updated on the test set. The resulting accuracies shall be compared in this case.