

Castle Labs

**Department of Operations Research and Financial Engineering**

**Princeton University**

**MOLTE**

**A modular, optimal learning testing environment**

Yingfei Wang

Warren B. Powell

Robert Schapire

March 24, 2014

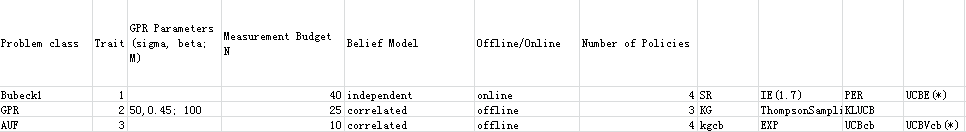
**Description**

MOLTE is a sequential design-of-experiments (stochastic optimization) testing environment for testing learning algorithms on a wide range of offline and online problems. The Matlab-based simulator allows the comparison of a number of learning policies (represented as a series of .m modules) in the context of a wide range of problems (each represented in its own .m module). The choice of problems and policies is guided through a spreadsheet-based interface. Users can follow the standard APIs to define a new problem class and new policy by writing a separate .m file.

**Construction**

Simulator.m compares the polices specified in the Excel spreadsheet for each problem class for numP=100 times (which can be modified in Simulator.m). Each time the simulator is run, it generates numTruth (which can be modified in Simulator.m) different sample paths, shared between all the policies, computes the value of the objective function for each sample path and then averages the numTruth trials as the expected final reward or the expected cumulative rewards. The user may select to evaluate policies using either an online (“bandit”) objective function, or an offline objective function (ranking and selection, stochastic search).

**Input Arguments**



Spreadsheet: an Excel file with each row a problem class with the specified policies under comparison. A possible spreadsheet is as follows:

For each problem, the following information has to be provided

**Problem class** is the name of a pre-coded problem with a specified truth function, the number of alternatives and a default noise level. If it is a user defined problem, the user should write a .m file with the same name as presented in this spreadsheet.

**Trait** specifies the characteristic of the problem class:

trait=1: test function with a fixed truth;

trait=2: Lookup table – Given prior belief about the function values

trait=3: Parameterized families - Given prior belief about the parameters of the function

**GPR Parameters** specifies the value of the parameter for Gaussian Process Regression. Specifically, the prior mean is drawn from N(0, \sqrt(sigma)), the covariance matrix is of the form sigma\*exp(-beta(x-x’)). M is the number of alternatives.



**Measurement Budget** specifies the time horizon of the decision making procedure.

**Belief Model** decides whether we are using independent or correlated beliefs for the policies which use a Bayesian belief model.

**Offline/Online** controls whether the objective is to maximize the expected final reward or the expected total rewards.

**Number of Policies** is the number of policies under comparison. This specifies the number of columns which contain the name of a policy to be tested, each represented in the corresponding .m file with the same name. If there are parentheses with a number after the name of the policy, it means setting the tunable parameter to the value specified in the parentheses. If there are parentheses with \*, it means tuning the parameters for this problem class and using the tuned value in the comparison: Otherwise use the default value (in fact some policies, e.g. KG and Kriging, do not have tunable parameters).

**Output Data and Figures**

All the data and figures are saved in a separate folder for each problem class. Within the folder of each problem class:

**objectiveFunction.mat** saves the value of the online or offline objective function achieved by each policy for each trial;

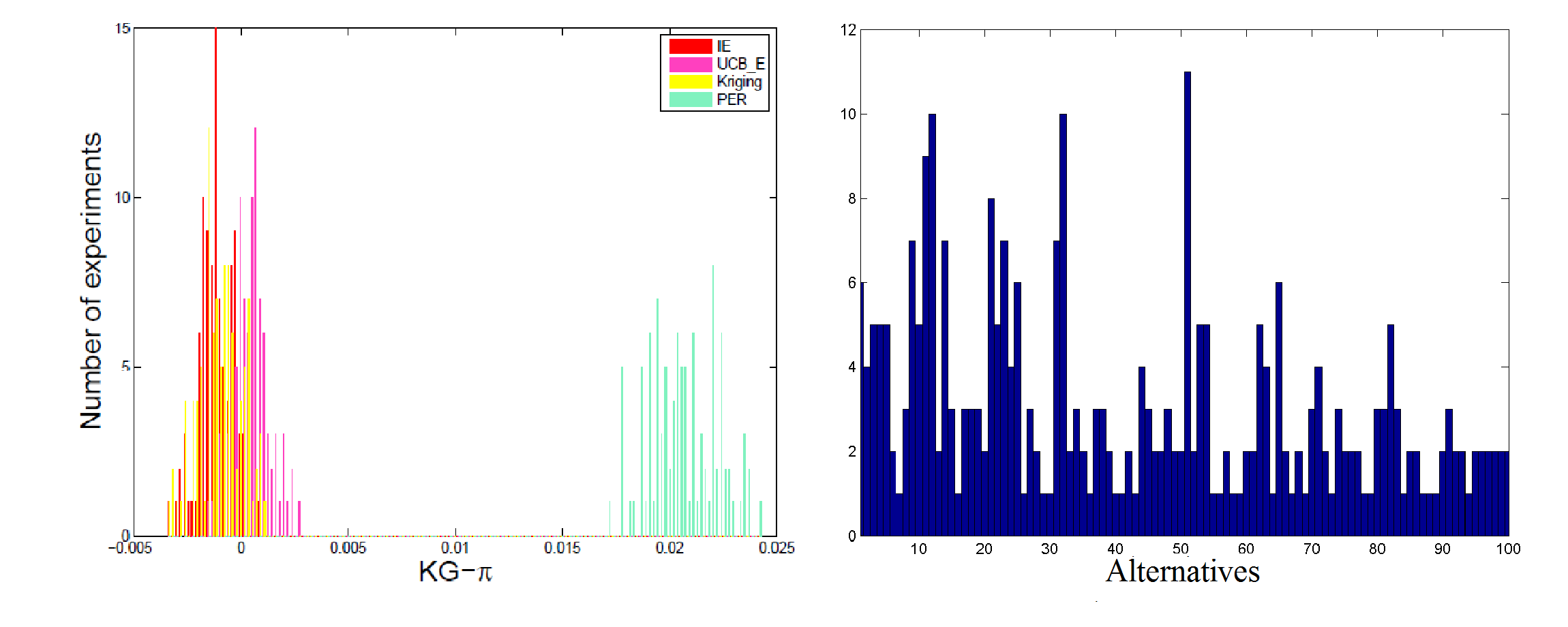
**choice.mat** saves the decisions made by each policy and the name of all policies;

**alpha.txt** saves the value of tunable parameter for each policy that requires tuning;

**offline\_hist.pdf** is the histogram for each policy describing the distribution of the expected final reward compared to the reward obtained by KG-type policy;

**online\_hist.pdf** is the histogram describing the distribution of the expected total reward; e.g. the following left figure is obtained for online Bubeck4. A distribution centered around a positive value implies the policy underperforms KG;

**histChoice.m** can read in the **choice.mat** and generate the distribution of the chosen alternatives for each policy and each trial. e,g, the right figure shows the frequency of choosing each of the 100 alternatives with a measurement budget of 300.



**Pre-coded Problem Classes**

**Synthetic test functions:**

* Bubeck1~Bubeck7
* Asymmetric Unimodular Functions with the parameter chosen to be 0.2,0.5 and 0.8 with high or medium noise level, respectively
* Rosenbrock function with additive noise
* Pinter's function with additive noise
* Goldstein function with additive noise
* Griewank function with additive noise
* Branin’s function with additive noise
* Axis parallel hyper-ellipsoid function with additive noise
* Rastrigin’s function with additive noise
* Ackley’s function with additive noise
* Six-hump camel back function with additive noise
* Easom function with additive noise

**Truth-From-Prior experiments:**

* Gaussian Process Regression

**Parameterized families**

* General AUF problems represented by one parameter drawn from U[0,1]

**Pre-coded Policies**

* Interval Estimation (IE) (can be used for correlated beliefs)
* Kriging (can be used for correlated beliefs)
* UCB (and a modified version UCBcb incorporating correlated beliefs)
* UCBNormal
* UCB-E (and a modified version UCBEcb incorporating correlated beliefs)
* UCB-V(and a modified version UCBVcb incorporating correlated beliefs)
* Bayes-UCB (can be used for correlated beliefs)
* KL-UCB
* Knowledge gradient policy for offline learning (can be used for correlated beliefs)
* Knowledge gradient for online learning (can be used for correlated beliefs)
* Successive rejects
* Thompson sampling (can be used for correlated beliefs)
* Pure exploration (can be used for correlated beliefs)
* Pure exploitation (can be used for correlated beliefs)