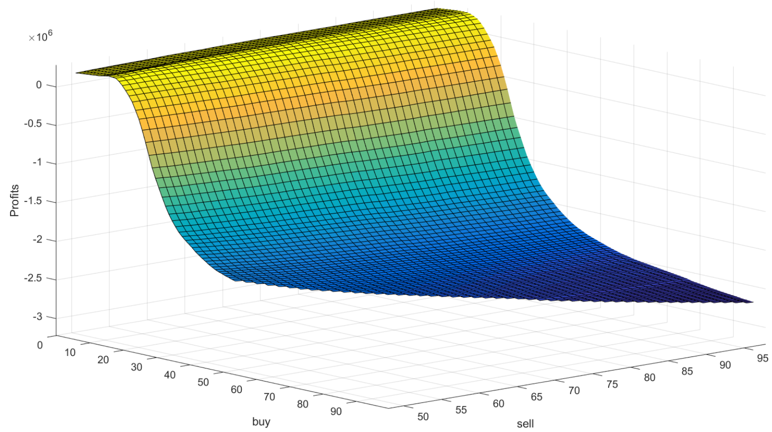
MOLTE-DB (Derivative Based)

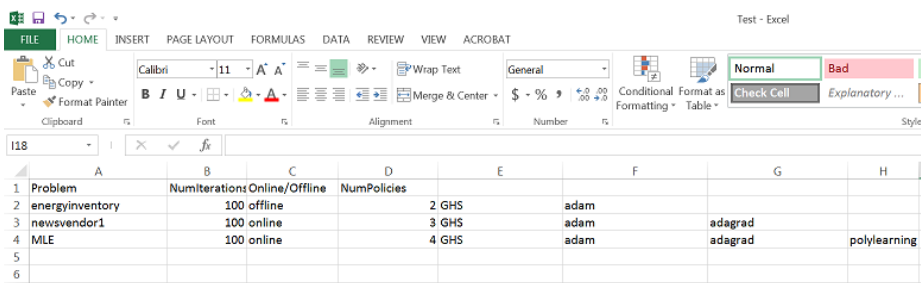
August 10, 2017



Description

MOLTE-DB is a Matlab derivative based stochastic optimization environment for testing a variety of stepsize policies on problems where the gradient can be computed or approximated. The simulator allows the comparison of stepwise policies (each represented in its own .m file) in the context of a range of problems (newsvendor, MLE, energy inventory). The user interacts with the interface through a Microsoft Excel spreadsheet, where problems and policies can be selected. Users can define new problems and new policies by creating a .m file.

Input Arguments



	A	B	C	D	E	F	G	H
1	Problem	NumIterations	Online/Offline	NumPolicies				
2	energyinventory	100	offline	2	GHS	adam		
3	newsvendor1	100	online	3	GHS	adam	adagrad	
4	MLE	100	online	4	GHS	adam	adagrad	polylearning
5								
6								

The user interacts with MOLTE-DB through an Excel spreadsheet called "DerivativeBased.xslx". The first column specifies the problem, second column specifies online or offline (cumulative or terminal reward), third column specifies the number of policies to test, and the next columns specify the names of the policies. The user can input parameters by typing in parentheses after the policy name. For example newsvendor(1,0.5,).

Supported Problem Classes

The optimum solutions for all these problem classes are obtained through the stochastic gradient algorithm: $\theta^{t+1} = \theta^t + \alpha \nabla F_\theta$

Newsvendor

Finding the optimal supply given a demand from an unknown random distribution and a desire to maximize profit (not overproduce such that supply is wasted or under-produce such that more demand could be fulfilled).

.m files

newsvendor: demand is normally distributed, price is much larger than cost
newsvendor1: demand normally distributed, price is close to cost
newsvendor2: demand exponentially distributed price is much larger than cost

newsvendor3: demand exponentially distributed price is close to cost

These files all output a vector of optimal supply estimates as well as vector of profits.

Maximum Likelihood Estimation (linear model)

Find the parameters θ for a linear model y = $\theta_1 x_1$ + $\theta_2 x_1$ + ... + $\theta_n x_n$ using stochastic gradient algorithm given a set of observation data. Note that for multiple parameters, our gradient is a vector of partial derivatives of our objective function with respect to each parameter.

.m files

MLE: one parameter
MLE1: two parameters

*BAKF is not supported for this problem.

Energy inventory policy

Given pricing data for battery storage, try to find a sell price, θ_s, and buy price, θ_b, such that profit is maximized. Program uses random restart to initialize starting point for stochastic gradient algorithm since the problem is nonconvex. The default number of times we randomly restart is 5. The largest profit out of the five different starting points is saved.
Versions of the program include one that gets prices from real pricing data (energyinventory) and a program that obtains prices from a jump diffusion process (energyinventoryp1). The jump diffusion process is given by the following formula:

.m files

energyinventory: Prices are obtained from real pricing data.
energyinventorp1: Prices are obtained from a jump diffusion process given by the following formula

$P^{t+1} = P^t + \Omega(P^t - P^{Energy}) + \quad \epsilon^{base} + I_t \epsilon^{jump}$

where $\epsilon^{base} \sim N(0, \sigma^2_{base}), \epsilon^{jump} \sim N(\mu_{jump}, \sigma^2_{jump})$, and I_t is 1 with some low probability α and zero with probability 1- α

*BAKF is not supported for this problem.

Output

Produces a plot of "profits" or in the case of MLE, the mean squared errors (MSE). The first policy entered in the spreadsheet is the benchmark. Bars to the right of zero are "better" than the benchmark and bars to the left are "worse" than the benchmark (values are calculated as profit - referenceprofit). For all problem classes, positive is "better".
Note that the parameter numPaths has been hardcoded into MOLTE-DB. For newsvendor and MLE problems, this parameter determines how many times a problem is simulated. If numPaths is 5, we will obtain 5 profits for each problem. For the sake of reasonably short runtimes, we only simulate the energy inventory policy one time for each stepsize policy.