Bayesian Optimization (BO), introduction: a first, simple, but complete example

Objectives: understanding the basic Bayesian optimization loop (acquire, update posterior, decide next sample, repeat) via an example.

Presentation in video: https://personales.upv.es/asala/YT/V/boloop1EN.html

Table of Contents

Motivation, preliminary ideas

Goal of BO: obtaining the global ***minimum*** of $f(x)$ via a "statistical model" of it (a Gaussian Process, actually).

Methodology:

1. Set a "prior" GP model (+ maybe some initial preexisting samples)
2. Decide which is the best "next x" to try, x_k.
3. Acquire samples (usually one by one, costly process), pairs $(x_k, y_k = f(x_k) + \epsilon_k)$.
4. Update the GP model (obtain the "posterior")
5. Evaluate termination criteria and go to step 2 if not met.

Step 2, detail: we associate to each x an "acquisition function" $a(x)$ related to how "good" the candidate sampling point is assumed to be, based on some statistical quantities. The x achieving "best" value of $a(x)$ is the one proposed to actually try in step 2.

We will use LOWER CONFIDENCE BOUND (2.5%) in this example; other options and refinements will be left for future videos.

Gaussian process example setup

True underlying function to optimize (unknown to BO):

```
TrueF=@(x) 0.9*sin(1.8*x+1.35)+.5*x.^2-.6+0.2.*x; %perfect
MaxSamples=12;
Data.priormean=@(x) zeros(size(x));
```

Prior: mean function and covariance kernel, confidence interval

```
Data.X=[]; Data.Y=[]; %To test "prior" model, empty data
Data.K=@K;
STDMeasureNoise=4e-2;
VarMeasureNoise=STDMeasureNoise^2;
Xtest=-2:(1/50):2; %uniform grid to test stuff
TrueData=TrueF(Xtest);
[TrueOptimum,TrueOptIdx]=min(TrueData);
TrueXOpt=Xtest(TrueOptIdx);
LL=length(Xtest);
[MeanPrior,CovMatrixPrior]=predictGP(Xtest,Data,VarMeasureNoise);
sg=sqrt(diag(CovMatrixPrior));
PredPrior=[MeanPrior-1.96*sg MeanPrior MeanPrior+1.96*sg];
```

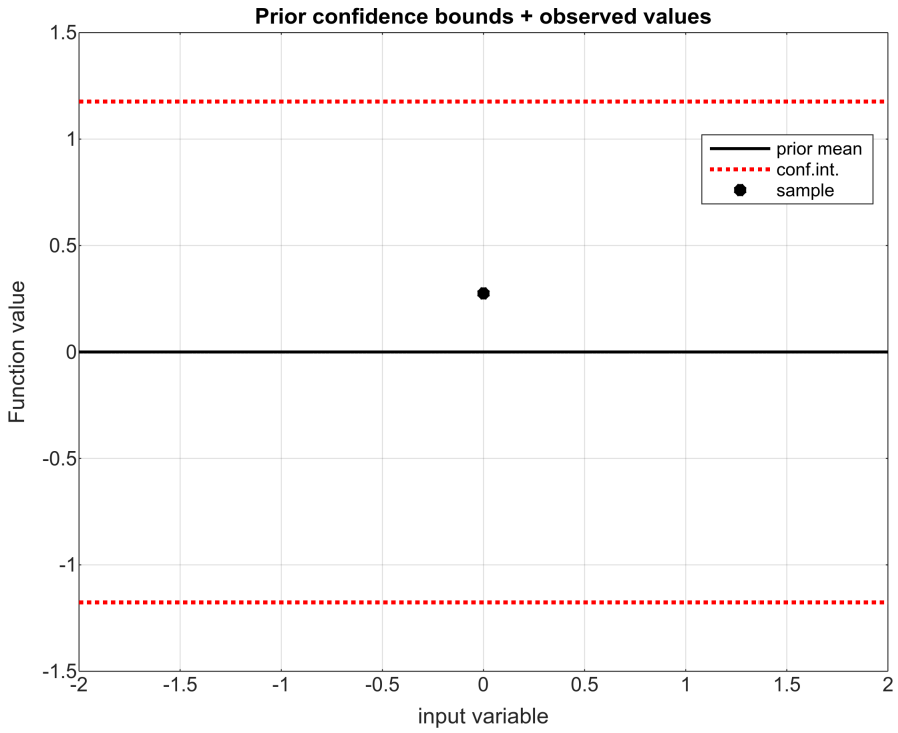
First sample

```
Data.X=0;
Data.Y=TrueF(Data.X)+randn(size(Data.X))*STDMeasureNoise;Data.Y
```

```
ans = 0.2749
```

```
plot(Xtest,PredPrior(:,2),'k',LineWidth=1.5), grid on, hold on
plot(Xtest,PredPrior(:,[1 3]),'r:',LineWidth=2)
plot(Data.X,Data.Y,'*k',LineWidth=3),
%plot(Xtest,TrueData,'-.',LineWidth=3,Color=[.4 .5 .2])
hold off
xlabel("input variable"), ylabel("Function value")
title("Prior confidence bounds + observed values")
legend("prior mean","conf.int.","","sample","true
F",Location="best")
```

```
Warning: Ignoring extra legend entries.
```



Bayesian optimization loop

```
ybestDataPLOT=[];
while length(Data.X)<MaxSamples
    [post_mean,post_var]=predictGP(Xtest,Data,VarMeasureNoise);
    [ybestData,idxbest]=min(Data.Y);
    ybest=ybestData
    ybestDataPLOT=[ybestDataPLOT ybest];
```

Note: minimising over a grid does not scale up to higher dimensions. Refinements are needed.

```
    sg=sqrt(diag(post_var));
    Pred=[post_mean-1.96*sg post_mean post_mean+1.96*sg];
    [min_lcb,lcb_idx]=min(post_mean-1.96*sg);

    AFname='LCB';
    NewSample=Xtest(lcb_idx)
    NewMeasure=TrueF(NewSample)+randn()*STDMeasureNoise
```

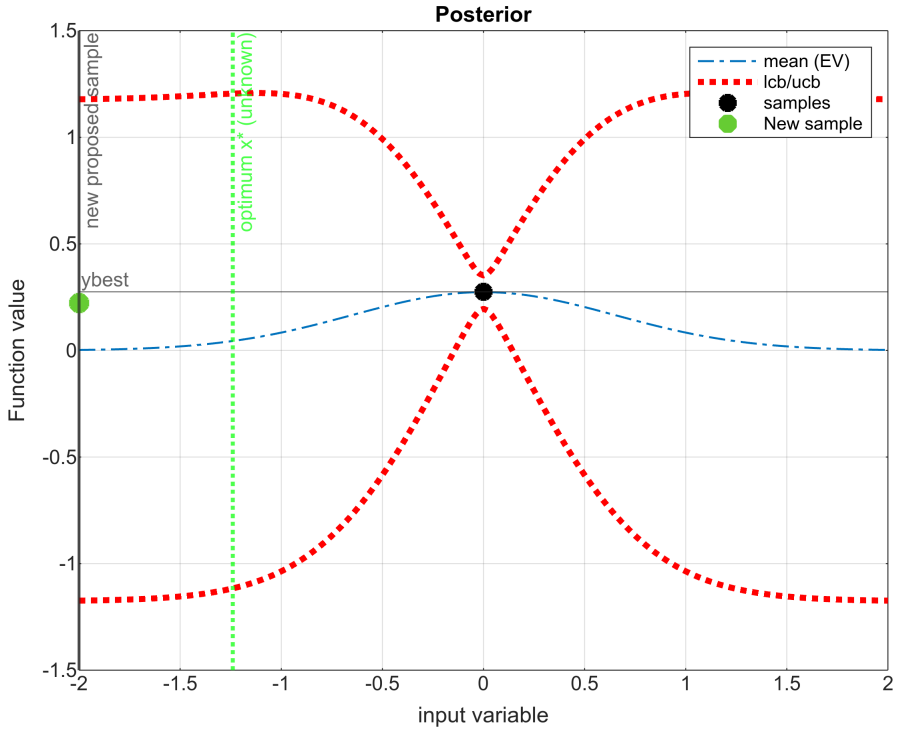
Cosmetic plotting code

```
    figure()
    plot(Xtest,Pred(:,2),"-.",LineWidth=1)
    hold on
    plot(Xtest,Pred(:,[1 3],:),'r:',LineWidth=3)
    plot(Data.X,Data.Y,'*k',LineWidth=3,MarkerSize=9);
    %plot(Xtest,TrueData,'-.',LineWidth=2,Color=[.4 .5 0.2])
```

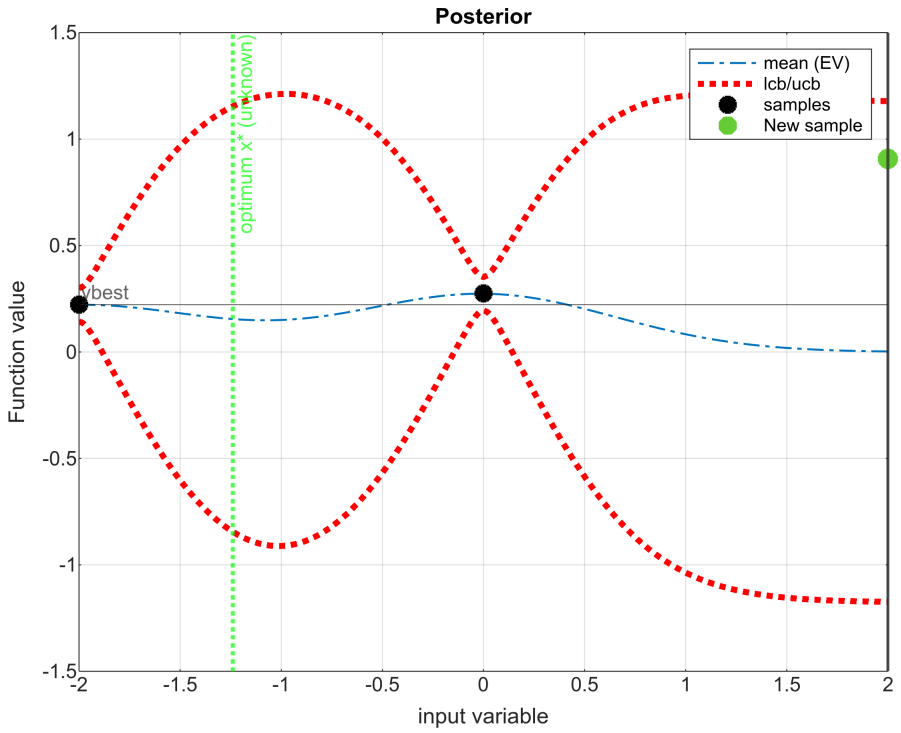
```
    plot(NewSample,NewMeasure,'*',Color=[.4 .8
.2],LineWidth=3.5,MarkerSize=10);
    hold off
    grid on, xlabel("input variable"), ylabel("Function value")
    title("Posterior")
    yline(ybest,label="ybest",LabelHorizontalAlignment="left")
    xline(NewSample,LineWidth=1.5,label=['new proposed sample'])
    xline(TrueXOpt,':g',LineWidth=2,label="optimum x* (unknown)")
    legend("mean (EV)","lcb/ucb","","samples","New sample")
    Data.X=[Data.X NewSample]; %for next iter
    Data.Y=[Data.Y NewMeasure]; %for next iter

end %end of Bayesian Optimization loopmedia
```

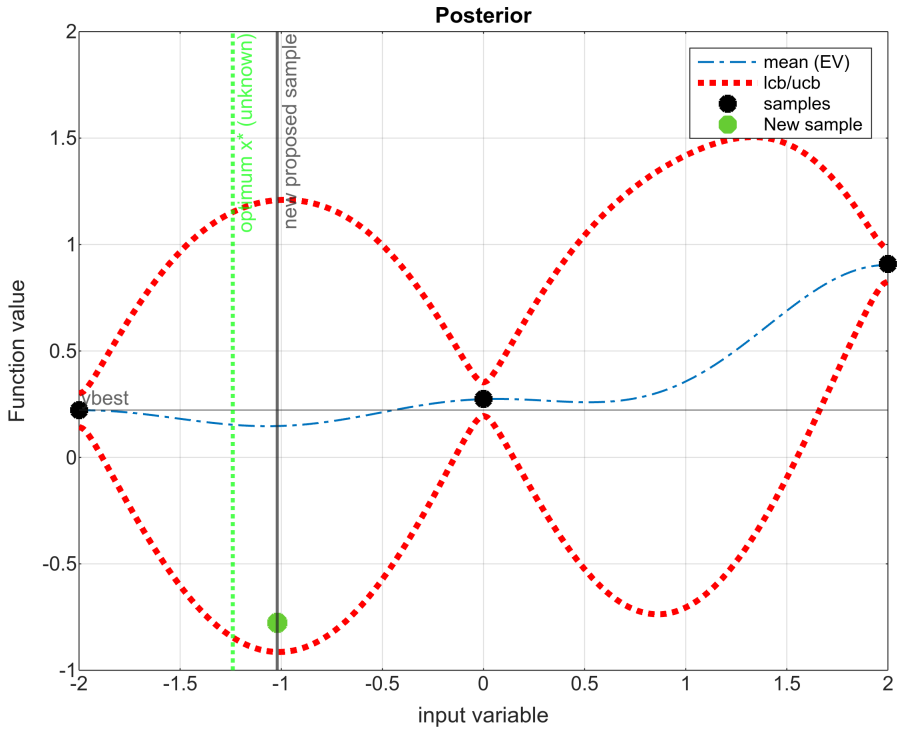
```
ybest = 0.2749
NewSample = -2
NewMeasure = 0.2224
```



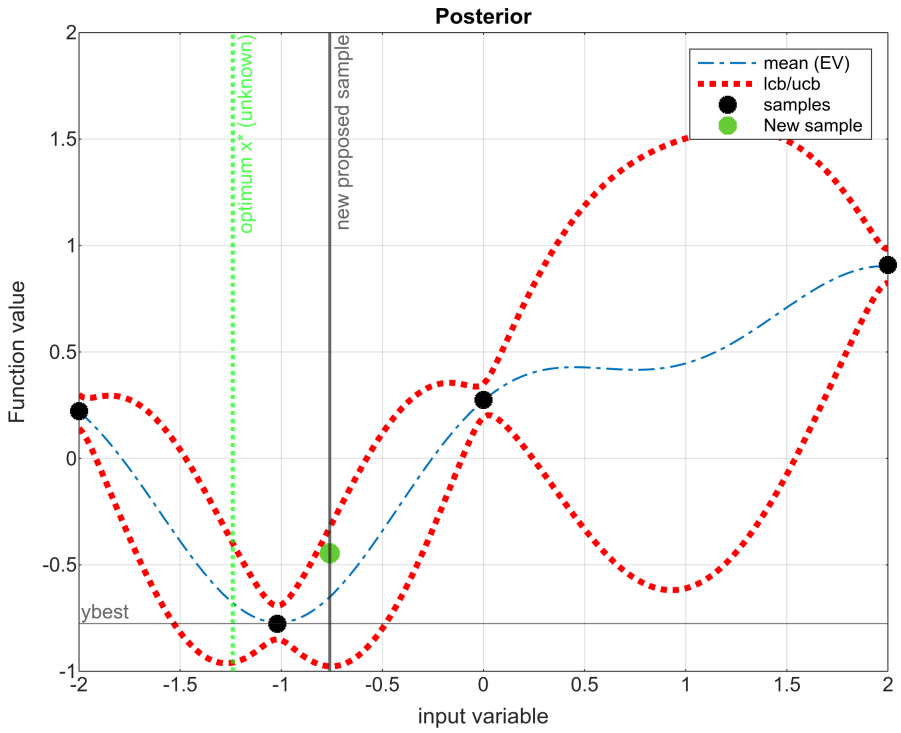

```
ybest = 0.2224
NewSample = 2
NewMeasure = 0.9077
```



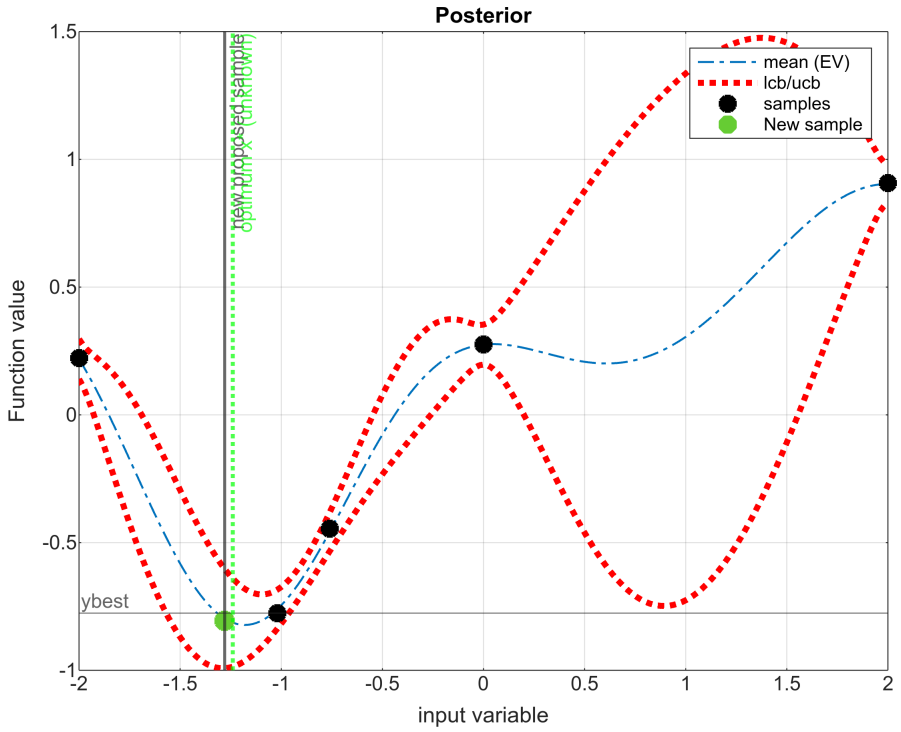

```
ybest = 0.2224
NewSample = -1.0200
NewMeasure = -0.7760
```



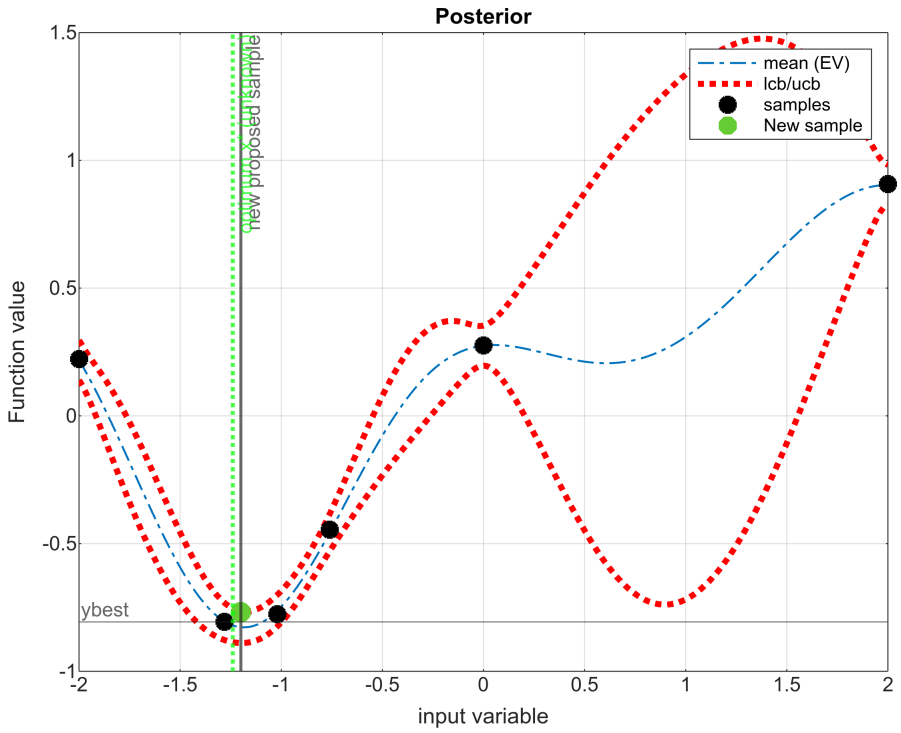

```
ybest = -0.7760
NewSample = -0.7600
NewMeasure = -0.4458
```



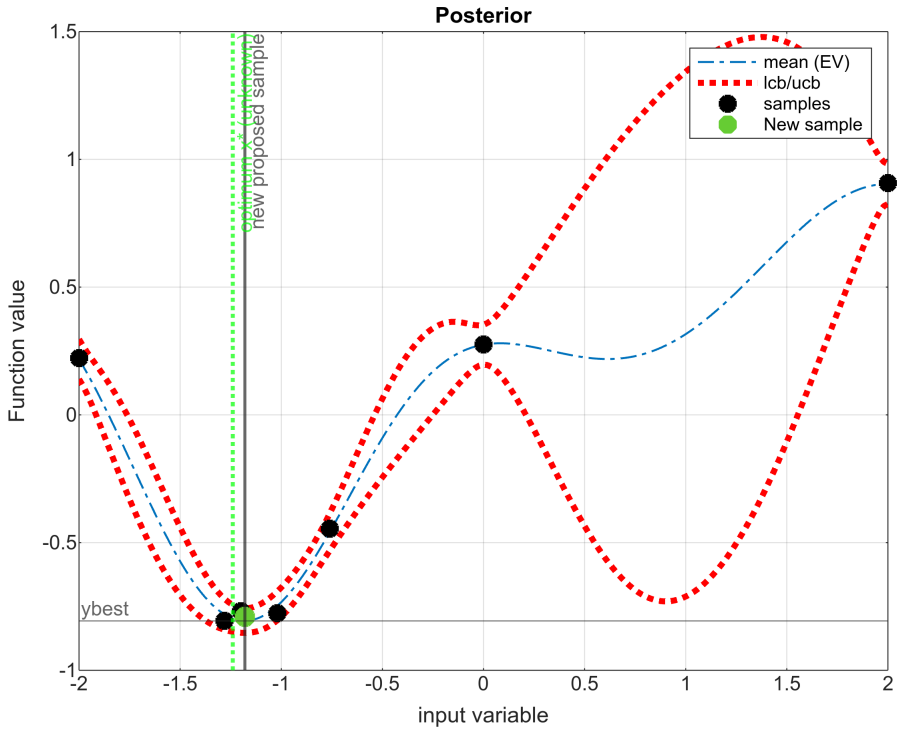

```
ybest = -0.7760
NewSample = -1.2800
NewMeasure = -0.8065
```



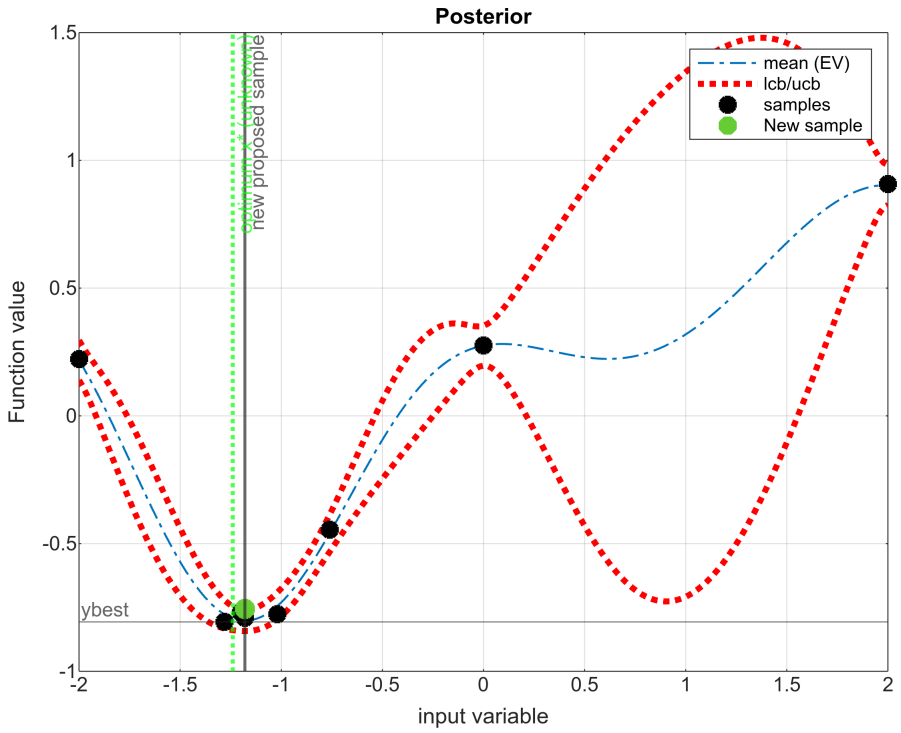

```
ybest = -0.8065
NewSample = -1.2000
NewMeasure = -0.7679
```



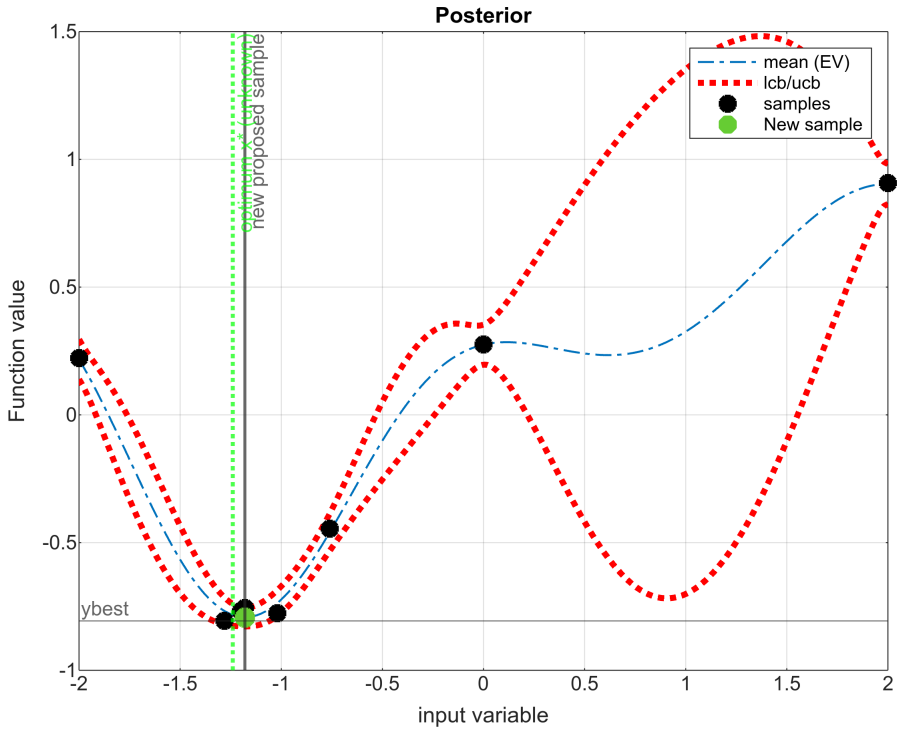

```
ybest = -0.8065
NewSample = -1.1800
NewMeasure = -0.7907
```



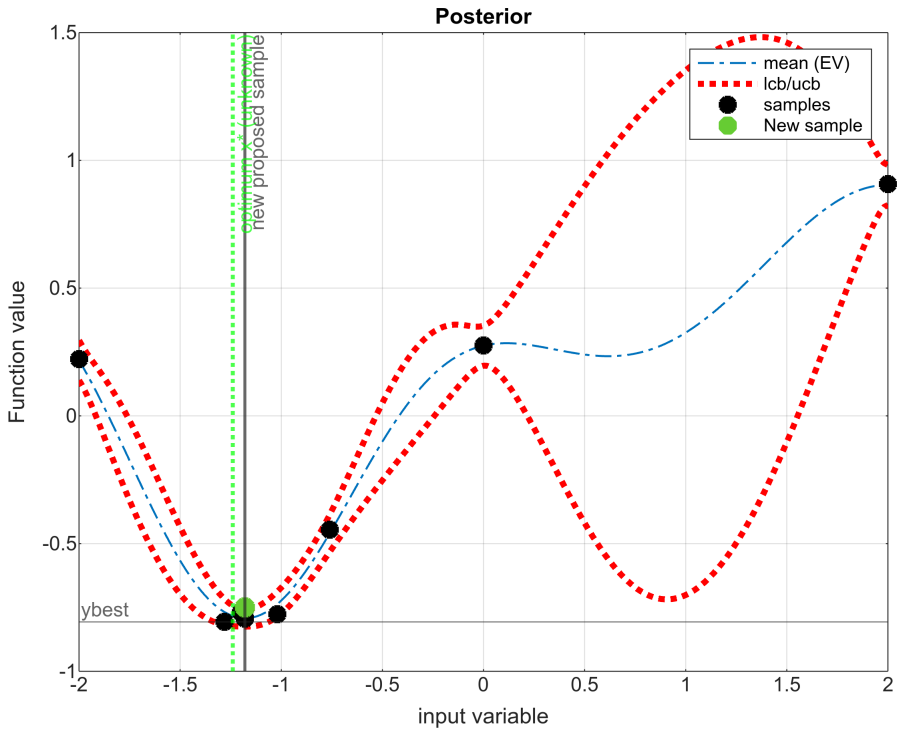

```
ybest = -0.8065
NewSample = -1.1800
NewMeasure = -0.7568
```




```
ybest = -0.8065
NewSample = -1.1800
NewMeasure = -0.7929
```

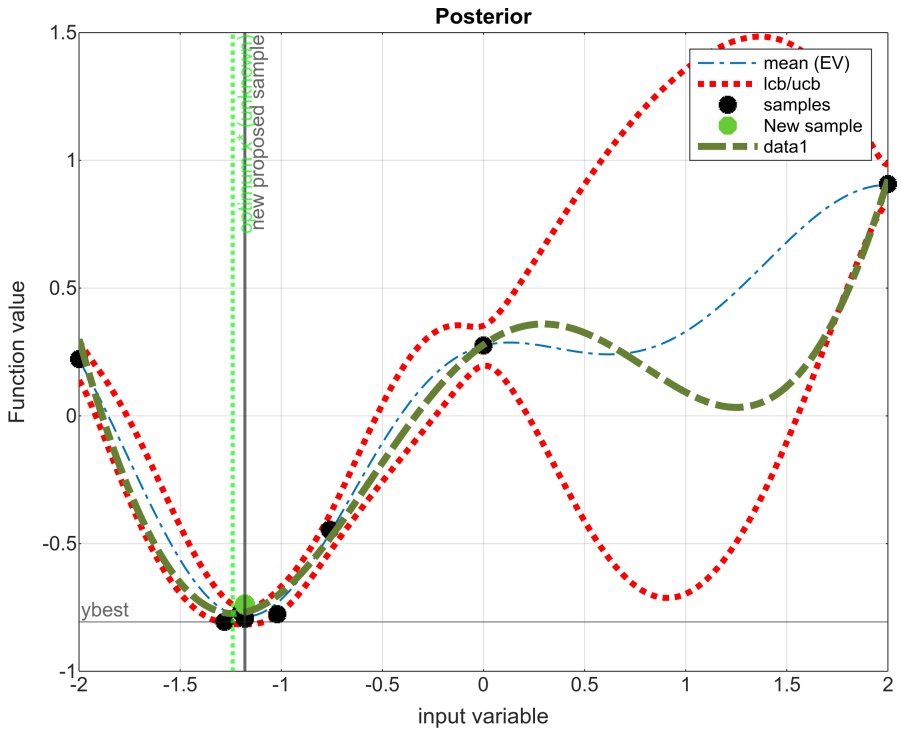



```
ybest = -0.8065
NewSample = -1.1800
NewMeasure = -0.7493
```



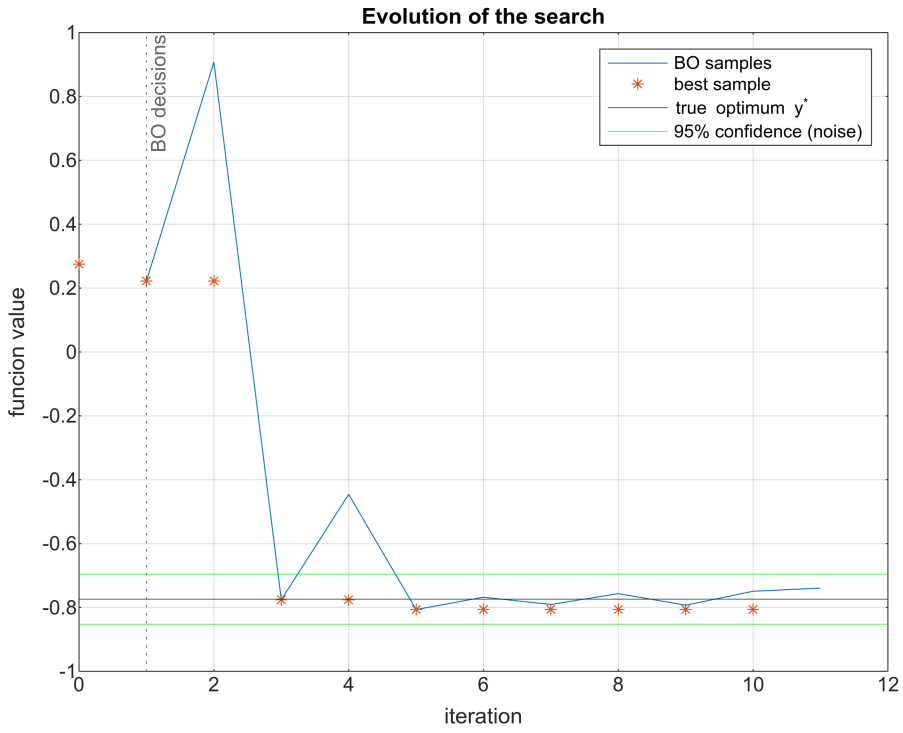

```
ybest = -0.8065
NewSample = -1.1800
NewMeasure = -0.7393
```

```
hold on
plot(Xtest,TrueData,'-.',LineWidth=3.5,Color=[.4 .5 0.2])
hold off
```



Progress plot:

```
plot(Data.Y(2:end)), hold on
plot(0:(MaxSamples-2), ybestDataPLOT,'*'),
hold off
yline(TrueOptimum)
yline(TrueOptimum+1.96*STDMeasureNoise,'g')
yline(TrueOptimum-1.96*STDMeasureNoise,'g'),
xline(1,':',label="BO decisions")
title("Evolution of the search"), grid on, xlabel("iteration"),
ylabel("funcion value")
legend("BO samples ","best sample ","true optimum y^*","95%
confidence (noise)")
```



Appendix: basic Gaussian process regression code

```
function [mu,Var]=predictGP(x1,Data,VarMeasureNoise)
priormean=Data.priormean;
K=@Data.K;
N=length(Data.X);
mu=priormean(x1)';
Var=K(x1,x1);
if(N>0)
    K1X=K(x1,Data.X);
    Gain=K1X/(K(Data.X,Data.X)+VarMeasureNoise*eye(N));
    mu=mu+Gain*(Data.Y-priormean(Data.X))';
    Var=Var-Gain*K1X';
    Var=0.5*(Var+Var'); %roundoff-errors, correct them
end
end
```

```
function km=K(x1,x2)
M=0.6; sg=0.65;%hyperparameters stddev x and y
kernel=@(x1,x2)   M^2*exp(-norm(x2-x1)^2/2/sg^2); %covariance
kernel function
N1=length(x1); N2=length(x2); %1D case
km=zeros(N1,N2);
for i=1:N1
    for j=1:N2
        km(i,j)=kernel(x1(:,i),x2(:,j)); %do it for all pairs
(x1_i,x2_j)
    end
end
end
```