

# Optimización bayesiana: ejemplo de funciones de adquisición, discusión

© 2024, Antonio Sala. Universitat Politecnica de Valencia, Spain. Todos los derechos reservados.

**Objetivos:** Analizar el significado de las "funciones de adquisición" populares en la literatura BO, basándose en un ejemplo sencillo.

## Presentaciones en vídeo:

<https://personales.upv.es/asala/YT/V/boPI.html>

<https://personales.upv.es/asala/YT/V/boEI.html>

## Tabla de Contenidos

Motivación, discusión preliminar.....	1
Inicialización del Proceso gaussiano ejemplo.....	2
Prior: función media y kernel, intervalo de confianza.....	2
Cargado de datos experimentales de observaciones "pasadas".....	3
Posterior y muestras del mismo.....	4
Muestras ejemplo del posterior.....	5
Información para la optimización bayesiana: distribución de puntos y valores óptimos.....	6
Ejemplos de uso de funciones de adquisición.....	7

## Motivación, discusión preliminar

**Objetivo de BO:** obtener el *mínimo* global de  $f(x)$  a través de un "modelo estadístico" del mismo (un **proceso gaussiano**, usualmente).

### Metodología:

1. Establecer un modelo de GP, el "prior".  $k = 0$  (o número de muestras iniciales disponibles)
2. Decide cuál es la mejor "siguiente x",  $x_k$  para probar.
3. Adquirir muestras (generalmente una por una, proceso costoso), pares  $(x_k, y_k = f(x_k) + \epsilon_k)$ .
4. Actualizar el modelo GP (obtener el "posterior")
5. Evaluar los criterios de terminación e ir al paso 2 si no se cumplen.

**Paso 4, detalle:** asociamos a cada  $x$  una "función de adquisición"  $a(x)$  relacionada con lo "bueno" que se supone que es el punto de muestreo candidato, basado en determinadas propiedades estadísticas. El "mejor" valor de  $a(x)$  es el que se propone probar en el paso 2.

*Idea básica:* evaluar  $a(x)$  debe ser mucho más rápido que evaluar  $f(x)$ , porque si no, usaríamos la optimización "determinista" estándar.

**Opciones populares para  $a(x)$  son:**

- **EV:** Expected value, valor esperado, minimizar
- **PI:** Probability of improvement, probabilidad de mejorar, a maximizar
- **EI:** Expected improvement, mejora esperada, a maximizar
- **LCB:** Lower confidence bound, cota inferior de intervalo de confianza, a minimizar

Estas son las que se discuten aquí con un ejemplo.

**Otras opciones:** existen "*Knowledge gradient*", "*Thompson sampling*", y "*Entropy search*". Se dejan para discusión futura en otros materiales.

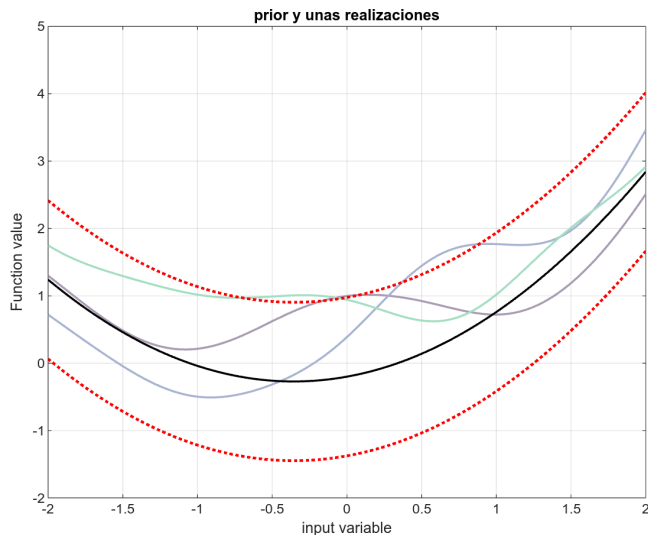
## Inicialización del Proceso gaussiano ejemplo

**Prior: función media y kernel, intervalo de confianza**

```
Data.X=[]; Data.Y=[]; %To test "prior" model, empty data
Data.K=@K;
Xtest=-2:(1/50):2; %uniform grid to test stuff
LL=length(Xtest)
```

```
LL = 201
```

```
[MeanPrior,CovMatrixPrior]=predictGP(Xtest,Data);
sg=sqrt(diag(CovMatrixPrior));
PredPrior=[MeanPrior-1.96*sg MeanPrior MeanPrior+1.96*sg];
NTrials=3;
FunctionRealization=mvnrnd(MeanPrior,CovMatrixPrior,NTrials);%clean
for i=1:NTrials
    col=[.5 .6 .65]+rand()*[.3 0 0]+rand()*[0 .3 0]+rand()*[0 0
0.2];
    plot(Xtest,FunctionRealization(i,:),Color=col,LineWidth=1.5),
hold on
end
plot(Xtest,PredPrior(:,2),'k',LineWidth=1.5), grid on
plot(Xtest,PredPrior(:,[1 3]),'r:',LineWidth=2), hold off
title("prior y unas realizaciones")
xlabel("input variable"), ylabel("Function value")
```



## Cargado de datos experimentales de observaciones "pasadas"

```
Data.X=[-0.8 -.2 1.45 1.95];
Data.Y=[0.75 1.2 0.48 1.75];
%Data.X=[-0.7 +1.65];
%Data.Y=[1.5 3.3];
%Data.Y=[0.5 -1.2];
```

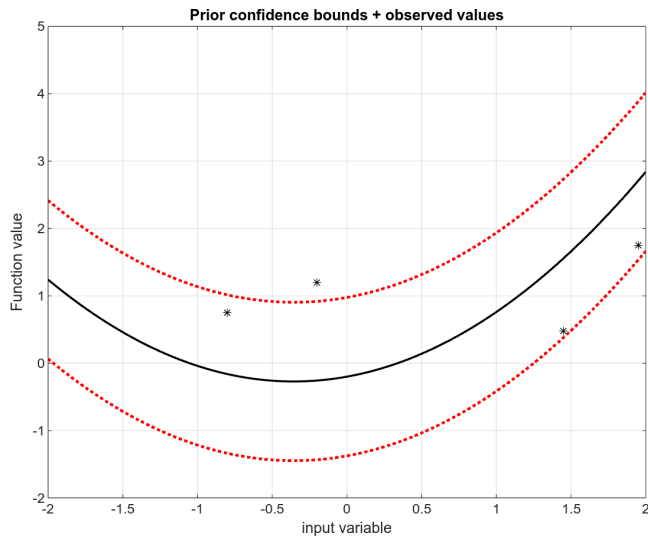
Escojamos la mejor muestra:

```
ybest=min(Data.Y);
```

**Nota:** con ruido de medida, quizás lo que deberíamos es tomar, en vez de Data:Y, la mejor "estimación media" del proceso gaussiano en Data.X para filtrarlo un poco. Esto evitará ciertos comportamientos "patológicos" en "probabilidad de mejora (PI)" y "mejora esperada (EI)". Se omite aquí la discusión por brevedad/simplicidad, ver siguientes vídeos.

Dibujemos:

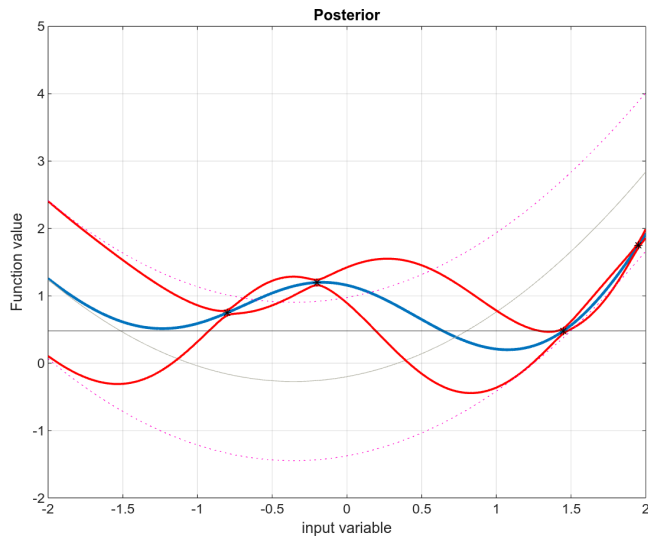
```
plot(Xtest,PredPrior(:,2),'k',LineWidth=1.5), grid on, hold on
plot(Xtest,PredPrior(:,[1 3]),'r:',LineWidth=2)
plot(Data.X,Data.Y,'*k'), hold off, xlabel("input variable"),
ylabel("Function value")
title("Prior confidence bounds + observed values")
```



¿Optimización de hiperparámetros?: todas las muestras están tan cerca/fuera de los límites de confianza del 95%... tal vez queramos jugar con los parámetros del kernel de covarianza. Bueno, recuérdalo para mejoras posteriores, pero déjalo de lado por el momento.

## Posterior y muestras del mismo

```
[post_mean,post_var]=predictGP(Xtest,Data);
sg=sqrt(diag(post_var));
Pred=[post_mean-1.96*sg post_mean post_mean+1.96*sg];
plot(Xtest,Pred(:,2),LineWidth=2)
hold on
plot(Xtest,PredPrior(:,2),Color=[.7 .7 .65])
plot(Xtest,Pred(:,[1 3],:),'r',LineWidth=1.5)
plot(Xtest,PredPrior(:,[1 3]),':',Color=[1 .0 .8])
plot(Data.X,Data.Y,'*k');
yline(ybest)
title("Posterior")
hold off, grid on, xlabel("input variable"), ylabel("Function
value")
```

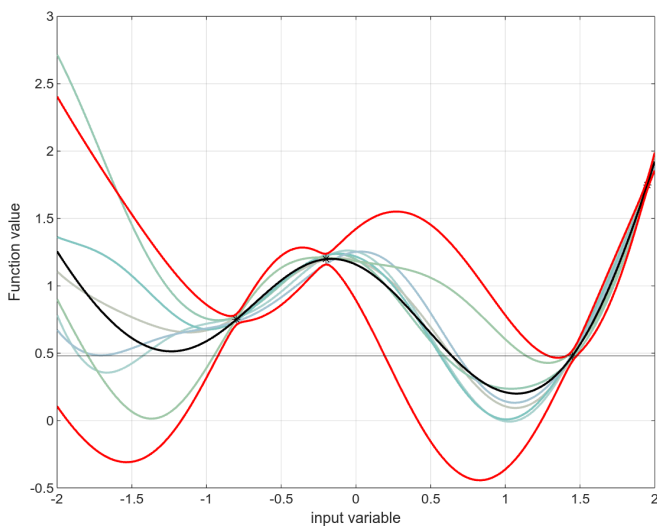


## Muestras ejemplo del posterior

```

figure()
NTrials=6;
FunctionRealization=mvnrnd(post_mean,post_var,NTrials);%clean
for i=1:NTrials
    col=[.5 .6 .65]+rand()*[.3 0 0]+rand()*[0 .3 0]+rand()*[0 0
0.2];
    plot(Xtest,FunctionRealization(i,:),Color=col,LineWidth=1.5),
hold on
end
hold on
plot(Data.X,Data.Y,'*k');
plot(Xtest,Pred(:,2),'k',LineWidth=1.5)
plot(Xtest,Pred(:,[1 3]),'r',LineWidth=1.5)
yline(ybest)
hold off, grid on, xlabel("input variable"), ylabel("Function
value")

```

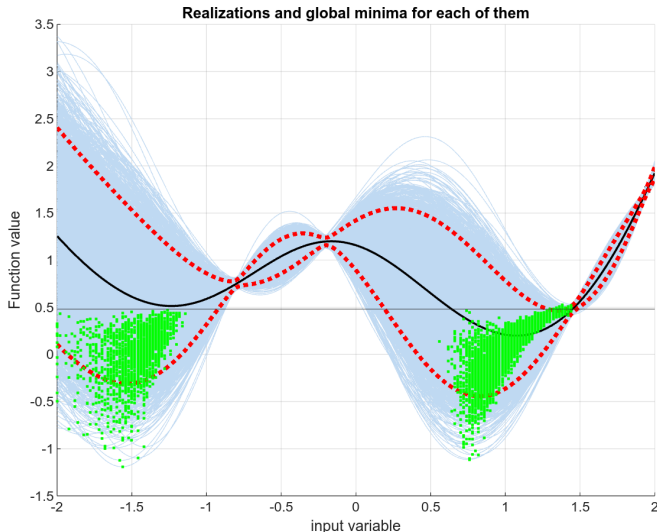


## Información para la optimización bayesiana: distribución de puntos y valores óptimos

```
figure()
hold on
plot(Data.X,Data.Y, '*k');
NTrials=8000;
FunctionRealizations=mvnrnd(post_mean,post_var,NTrials);
size(FunctionRealizations)
```

```
ans = 1x2
      8000      201
```

```
[fval,xmn]=min(FunctionRealizations,[],2); %min of each trial,
i.e., min of each ROW.
plot(Xtest,FunctionRealizations,Color=[.75 .85 .95])
plot(Xtest,Pred(:,2), 'k',LineWidth=1.5)
plot(Xtest,Pred(:, [1 3]), ':r',LineWidth=3)
plot(Xtest(xmn),fval, '.g')
hold off, grid on
yline(ybest)
title("Realizations and global minima for each of them")
xlabel("input variable"), ylabel("Function value")
```

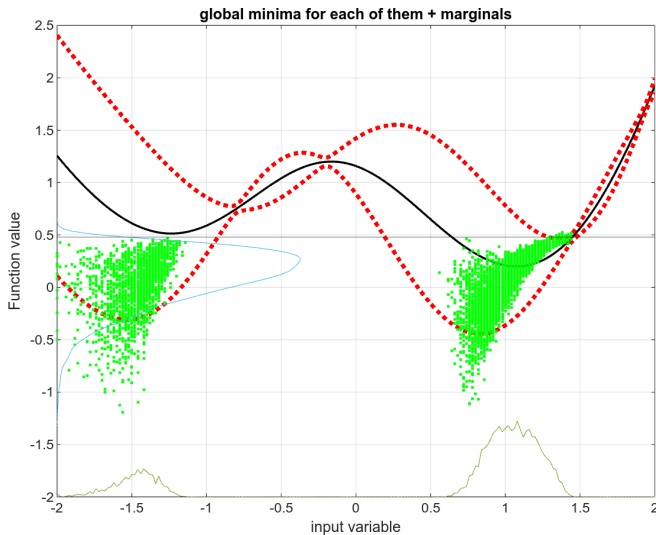


```
plot(Xtest,Pred(:,2), 'k',LineWidth=1.5)
hold on
plot(Xtest,Pred(:, [1 3]), ':r',LineWidth=3)
plot(Xtest(xmn),fval, '.g')
grid on
title("global minima for each of them + marginals")
xlabel("input variable"), ylabel("Function value")
histlp=zeros(LL,1);
```

```

for k=1:LL
    hist1p(k)=length(find(xmn==k));
end
Pmin=hist1p/sum(hist1p);
plot(Xtest,Pmin*20-2)
[pro,ff]=ksdensity(fval);
plot(pro-2,ff);
yline(ybest)
grid on, hold off

```



## Ejemplos de uso de funciones de adquisición

```

yn=@(mean,var,y) (y-mean)/sqrt(var); %Normalize to standard normal
(utility)

```

- Claramente, el valor esperado **EV** (media a posteriori) puede ser lo más fácil de entender... pero tenemos otras opciones.
- **PI**, probability of improvement, probabilidad de mejora:

```

PI=@(m,v,level) normcdf(yn(m,v,level)); %prob of. standard normal
distribution below "level"

```

- **EI**, expected improvement, mejora esperada (es conceptualmente mejor que el "valor esperado" si no hay "coste" por tener "mala suerte", es decir, probar un punto que no funciona bien)

```

EIN=@(c) -normpdf(c)-c*normcdf(c);
EI=@(m,v,ybest) EIN(yn(m,v,ybest))*sqrt(v);

```

- Umbral de confianza inferior (bueno, lo graficamos arriba y lo repetiremos a continuación).

```
PIplt=zeros(1,LL); EIplt=zeros(1,LL); %for graphics
```

Para todos los 201 puntos de test

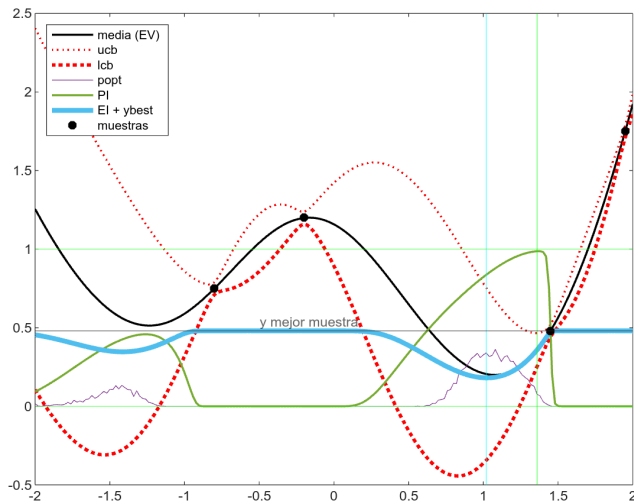
```
for i=1:LL
    PIplt(i)=PI(post_mean(i),post_var(i,i),ybest);
    EIplt(i)=EI(post_mean(i),post_var(i,i),ybest);
end
```

La mejor abscisa de cada propuesta de adquisición:

```
[min_ev,ev_idx]=min(post_mean);
[max_pi,pi_idx]=max(PIplt); %Maximum probability of improvement
[min_ei,ei_idx]=min(EIplt);
[min_ucb,ucb_idx]=min(Pred(:,3));
[min_lcb,lcb_idx]=min(Pred(:,1));
```

Representemos en gráfica:

```
figure()
plot(Xtest,Pred(:,2),'k',LineWidth=1.5), hold on
plot(Xtest,Pred(:,3),'r',LineWidth=1.5)
plot(Xtest,Pred(:,1),'r',LineWidth=2.5)
plot(Xtest,Pmin*10), %grid on
plot(Xtest,PIplt,LineWidth=1.5),
plot(Xtest,EIplt+ybest,LineWidth=3.5),
plot(Data.X,Data.Y,'*k',LineWidth=2.5)
xline(Xtest(pi_idx),'g')
xline(Xtest(ei_idx),'c')
%xline(Xtest(ucb_idx),'r')
%xline(Xtest(lcb_idx),'r')
%xline(Xtest(ev_idx),'k')
hold off
yline(ybest,label="y mejor muestra",LabelHorizontalAlignment="center")
yline(1,'g'), yline(0,'g')
legend("media (EV)", "ucb", "lcb", "popt", "PI", "EI + ybest", "muestras", Location="northwest")
```



```

function [mu,Var]=predictGP(x1,Data)
arguments
    x1
    Data
end
priormean=@(x) 0.56*x.^2+0.4*x-.2;
lambda=2e-2^2; %measurement noise variance
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)+lambda*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=0.64;%hyperparameters stddev x and y %PI was 0.65
kernel=@(x1,x2) M^2*exp(-norm(x2-x1)^2/2/sg^2);
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));
    end
end
end
end

```

