

# Optimización bayesiana (motivación): información probabilística sobre el punto óptimo (implícitamente) disponible en un proceso gaussiano

© 2025, Antonio Sala. Universitat Politècnica de València, Spain. Todos los derechos reservados.

**Objetivos:** valorar la información sobre dónde está el mínimo global de un GP (posterior a unos datos), y sobre cuál es el valor que alcanza.

**Presentación en vídeo:**

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

## Tabla de Contenidos

Configuración del proceso gaussiano.....	1
Media a priori y kernel, intervalo de confianza.....	1
Cargar con datos experimentales de observaciones pasadas.....	2
Computación y realizaciones del posterior.....	3
Información para la optimización bayesiana: probabilidad de puntos óptimos y valores óptimos.....	4
Gráficas de probabilidad aproximadas.....	5
Apéndice (funciones auxiliares GP).....	7

## Configuración del proceso gaussiano

### Media a priori 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
```

```
[Mean,CovMatrix]=predictGP(Xtest,Data);
size(CovMatrix)
```

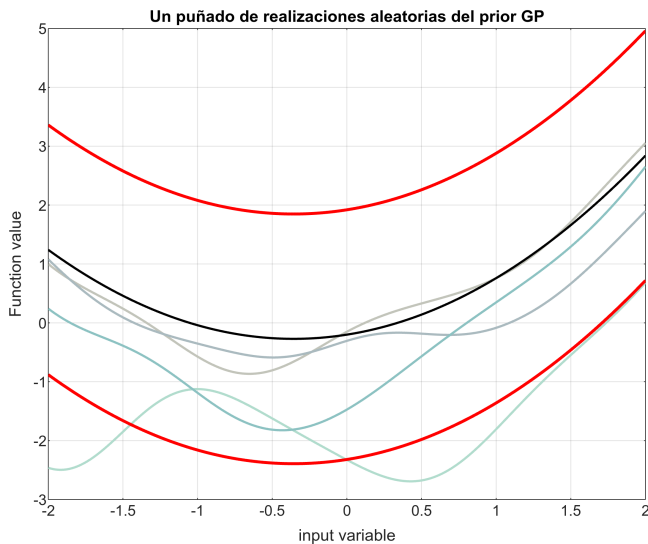
```
ans = 1x2
      201    201
```

```
sg=sqrt(diag(CovMatrix)); %standard deviation at each test point
PredPrior=[Mean-1.96*sg Mean Mean+1.96*sg]; %to plot conf. intervals
NTrials=4;
FunctionRealization=mvnrnd(Mean,CovMatrix,NTrials);
for i=1:NTrials
    if(NTrials<5)
        LW=1.5; %Line width, cosmetic
    else
        LW=1; %Line width, cosmetic
    end
```

```

col=[.5 .6 .65]+rand()*[.3 0 0]+rand()*[0 .3 0]+rand()*[0 0
0.2]; %color, cosmetic
plot(Xtest,FunctionRealization(i,:),Color=col,LineWidth=LW),
hold on
end
plot(Xtest,PredPrior(:,2),'k',LineWidth=1.5), grid on
plot(Xtest,PredPrior(:,[1 3]),'r',LineWidth=2), hold off
title("Un puñado de realizaciones aleatorias del prior GP")
xlabel("input variable"), ylabel("Function value")

```

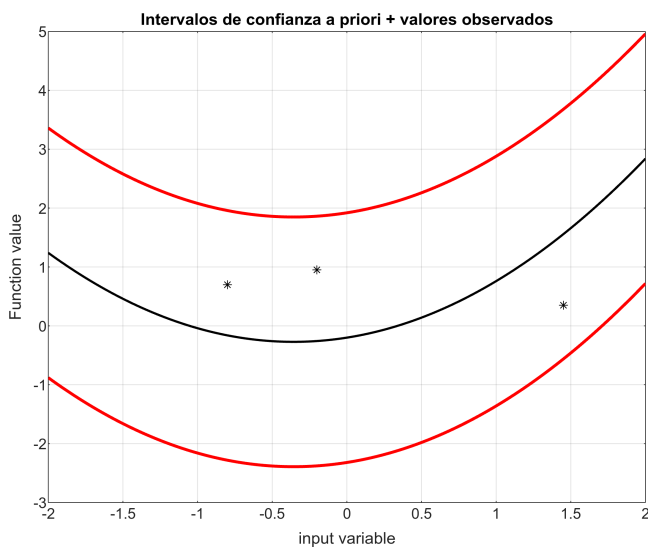


## Cargar con datos experimentales de observaciones pasadas

```

Data.X=[-0.8 -.2 1.45];
Data.Y=[0.7 0.95 0.35];
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("Intervalos de confianza a priori + valores observados")

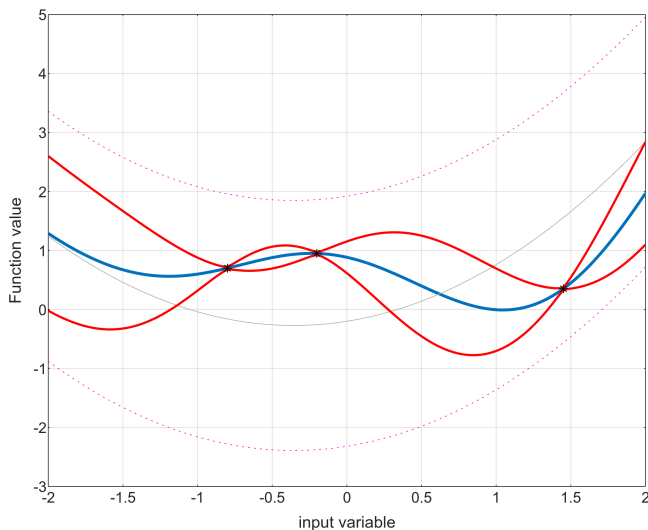
```



**Optimización de hiperparámetros?:** tal vez deseemos jugar con los parámetros del kernel de autocovarianza. Bueno, lo recordaremos para mejoras posteriores, pero olvídale por el momento.

## Computación y realizaciones del posterior

```
[mm, sss]=predictGP(Xtest,Data);
sg=sqrt(diag(sss)); %standard deviation at each test point
Pred=[mm-1.96*sg mm mm+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');
hold off, grid on, xlabel("input variable"), ylabel("Function
value")
```

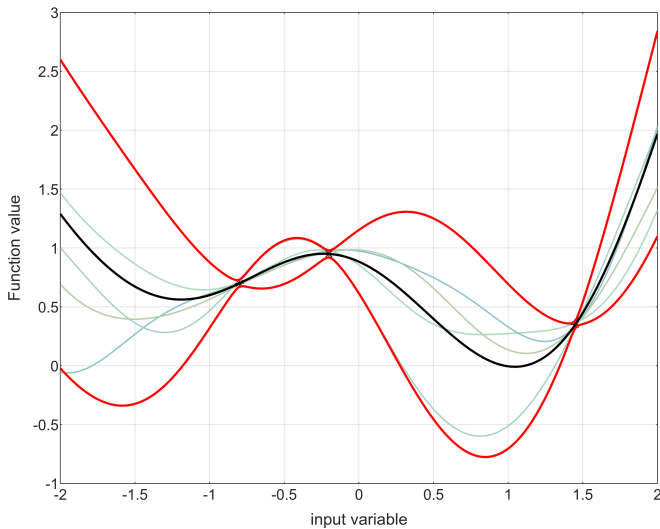


```
figure()
NTrials=4;
FunctionRealization=mvnrnd(mm, sss, NTrials); %clean
for i=1:NTrials
    if(NTrials<4)
        LW=1.5;
    else
        LW=1;
    end
    col=[.5 .6 .65]+rand()*[.3 0 0]+rand()*[0 .3 0]+rand()*[0 0
0.2];
    plot(Xtest,FunctionRealization(i,:),Color=col,LineWidth=LW),
hold on
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)
hold off, grid on, xlabel("input variable"), ylabel("Function
value")

```



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

Las pruebas experimentales son "costosas". Podemos hacer lo que queramos "in silico" (simulación, ver más abajo) pero obtener una muestra "real" requiere tiempo en pruebas de laboratorio, horas de mano de obra, materias primas, etc. Así que deseamos que nuestro "modelo estadístico" sobre cómo se comporta la función para guiarnos hacia la obtención de algo cercano al valor óptimo real con un número reducido de muestras experimentales.

Digamos que deseamos realizar la siguiente muestra en un punto que "es muy probable que sea óptimo", o en un punto que sea "altamente informativo sobre la ubicación del valor óptimo", o variaciones de eso. Dependiendo de los datos disponibles y presupuesto de muestras a realizar, podríamos ser más "arriesgados" (exploratorios) o menos.

Tiramos los dados con el GP y encontramos dónde está el mínimo global cada una de las realizaciones:

```

figure()
hold on
plot(Data.X,Data.Y,'*k');
NTrials=4500;
FunctionRealizations=mvnrnd(mm,sss,NTrials);
size(FunctionRealizations)

```

```

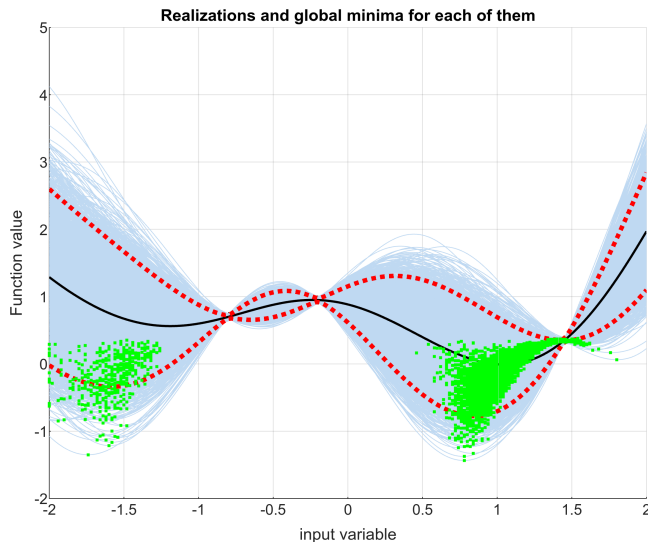
ans = 1x2
      4500      201

```

```

[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
title("Realizations and global minima for each of them")
xlabel("input variable"), ylabel("Function value")

```



## Gráficas de probabilidad aproximadas

```

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 realization + marginals")
xlabel("input variable"), ylabel("Function value")

```

### Histograma de entrada:

```

hist1p=zeros(LL,1);
for k=1:LL
    hist1p(k)=length(find(xmn==k));
end
Pmin=hist1p/sum(hist1p);
plot(Xtest,Pmin*20-2)

```

### Histograma de salida:

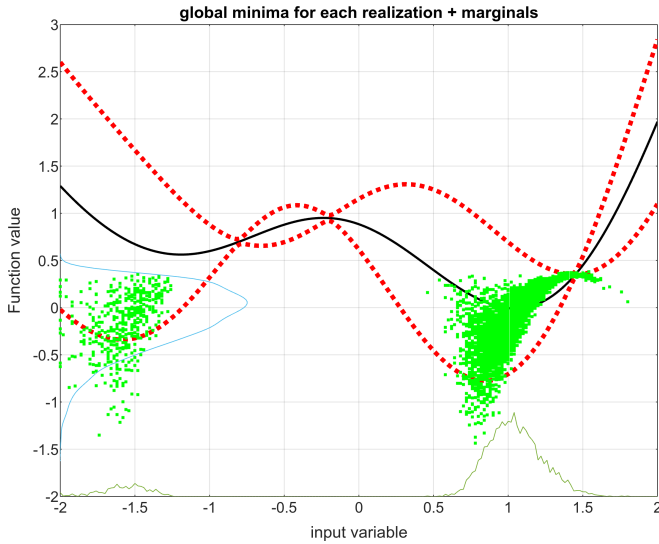
```

[Ncounts,edges]=histcounts(fval,Normalization="pdf");

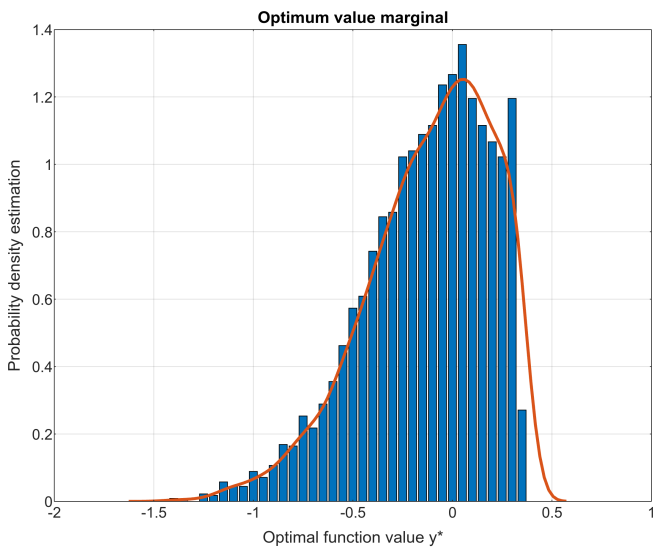
```

Histograma suavizado mediante "kernel density estimation" (no tiene nada que ver con el kernel de covarianza del GP):

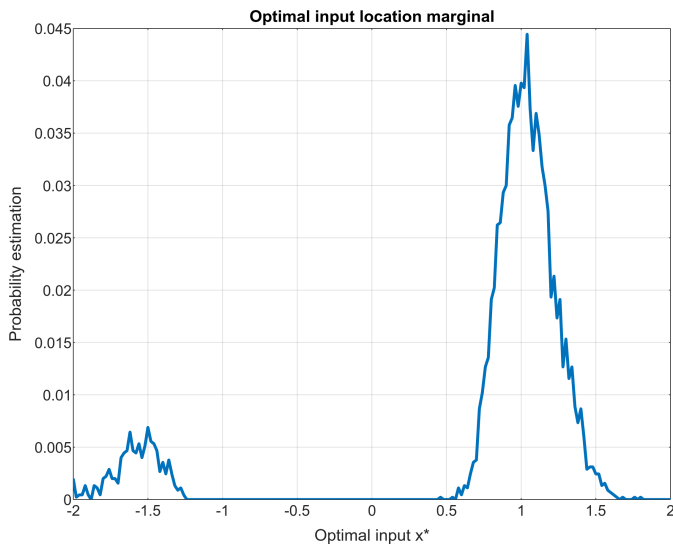
```
[pro,ff]=ksdensity(fval);
plot(pro-2,ff);
grid on, hold off
```



```
bar(edges(1:(end-1)),Ncounts), hold on
plot(ff,pro,LineWidth=2), xlabel("Optimal function value
y*"),ylabel("Probability density estimation"), grid on,
title("Optimum value marginal")
hold off
```



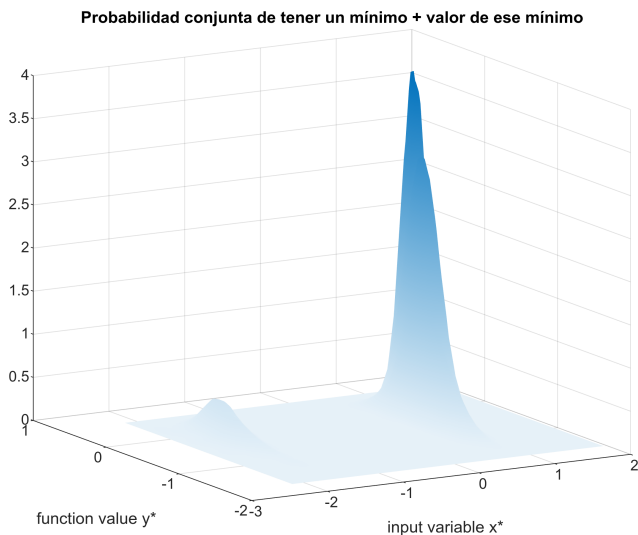
```
plot(Xtest,Pmin,LineWidth=2), xlabel('Optimal input x*'),
ylabel("Probability estimation")
hold off, grid on, title("Optimal input location marginal")
```



\*La estimación de la "densidad" de probabilidad requeriría dividir por el intervalo de separación entre puntos del mallado Xtest, proporcional al gráfico que vemos.

### Gráfico de probabilidad conjunta

```
ksdensity([Xtest(xmn)' fval]);
shading interp
view([-30 15])
colormap sky
xlabel("input variable x*"), ylabel("function value y*")
title("Probabilidad conjunta de tener un mínimo + valor de ese mínimo")
```



### Apéndice (funciones auxiliares GP)

```
function [mu, Var]=predictGP(x1, Data)
arguments
    x1
```

```

    Data
end
priormean=@(x) 0.56*x.^2+0.4*x-.2;
lambda=1.5e-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')+1e-15*eye(size(Var,1)); %round-off and
numerical issues just in case
end
end

function km=K(x1,x2)
M=0.6; sg=0.65;%hyperparameters: correlation distance sg (x) and
std dev (y)
kernel=@(x1,x2) M^2*exp(-norm(x2-x1)^2/2/sg^2)+0.9^2; %squared
exp.+some uncertainty on constant mean
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)); %we form the matrix
    end
end
end
end

```