Institución Universitaria
Reacreditada en Alta Calidad

Optimización

Black-box Optimization
Optimización Bayesiana

Docente: Cristian Guarnizo Lemus

Contenido

1. Introducción.
2. Procesos Gaussianos.
3. Técnicas para NLPs.

Objetivo de la presentación

- Entender los conceptos básicos de la optimización de caja negra (black-box).
- Entender el funcionamiento básico de la optimización Bayesiana.
- Cuando es útil emplear este tipo de optimización.

Black-box Optimization

Función Objetivo

Parámetros Optimizados

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in R^D} f(\boldsymbol{x})$$

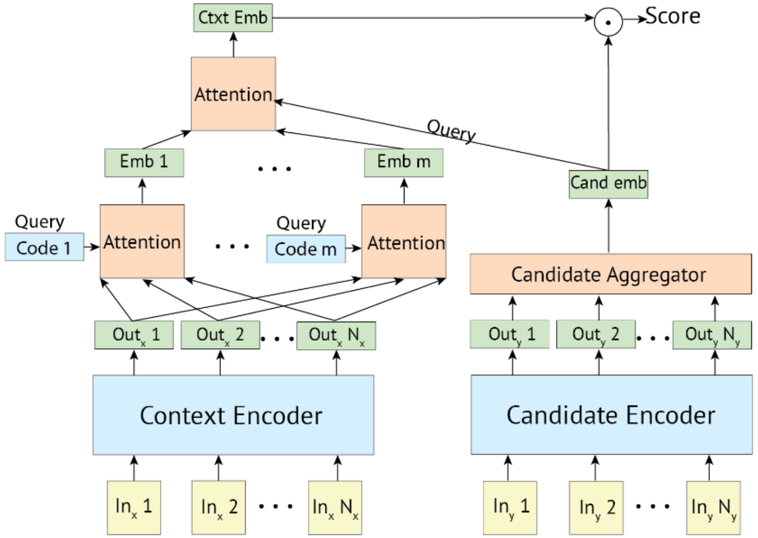
Parámetros a Optimizar

$$f(\boldsymbol{x}) = ?$$

Tipos de funciones objetivos

Un solo mínimo (p.e. funciones convexas)	Múltiples mínimos (optimización global)
De primer orden (podemos calcular gradientes)	Orden cero (sin gradientes)
Sin ruido (una evaluación repetida entrega el mismo resultado)	Estocástico (evaluaciones repetidas entregan diferentes resultados)
Evaluación barata (se permiten una cantidad de evaluaciones infinitas)	Evaluación costosa (limitado a decenas o cientos de evaluaciones)
Fácil de resolver (p.e. con gradientes)	Difícil de Optimizar

Tipos de funciones objetivos



Optimización de Hiperparametros

- Los modelos de aprendizaje automático están creciendo más y más complejos (Deep learning).
- Modelos de aprendizaje profundo modernos tienen docenas de parámetros que requieren ser optimizados (tasas de aprendizaje, numero de capas, tamaño del lote). (p.e. Optuna)
- Para alcanzar resultados del estado del arte, encontrar buenos hiperparametros es fundamental.
- Inclusive para expertos encontrar buenos hiperparametros puede ser difícil y consumir mucho tiempo.

Métodos de optimización tradicionales

- Sintonización manual (requiere del experto).
- Grid search (no escala bien a espacio de parámetros grandes).
- Random Search (mejor que Grid, pero aun así requiere muchas evaluaciones).
- Gradiente descendente (solo factible para funciones de primer orden).
- Búsqueda Heuristica (requiere de miles de evaluaciones)

Para docenas de parámetros, con correlaciones complejas, y evaluaciones costosas estos métodos se vuelven imprácticos.

Intuición detrás de la Optimización Bayesiana

Muchos optimizadores capturan solamente la información local de la función objetivo

$$\boldsymbol{x}^{(t+1)} = g\left(\boldsymbol{x}^{(t)}, f\left(\boldsymbol{x}^{(t)}\right)\right) \qquad \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha g\left(\nabla f\left(\boldsymbol{x}^{(t)}\right)\right)$$

Cómo podemos usar toda la información (evaluaciones) recolectadas hasta ahora para tomar decisiones más informadas, por ende mejorando eficiencia de los datos?

$$\boldsymbol{x}^{(t+1)} = g(D) \qquad D = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}, i = 1, \dots, n$$

Como hacer esto en la practica?

Podemos construir un modelo sustituto (Surrogate) $\tilde{f}(x_i)\Big|_D \sim f(x)$

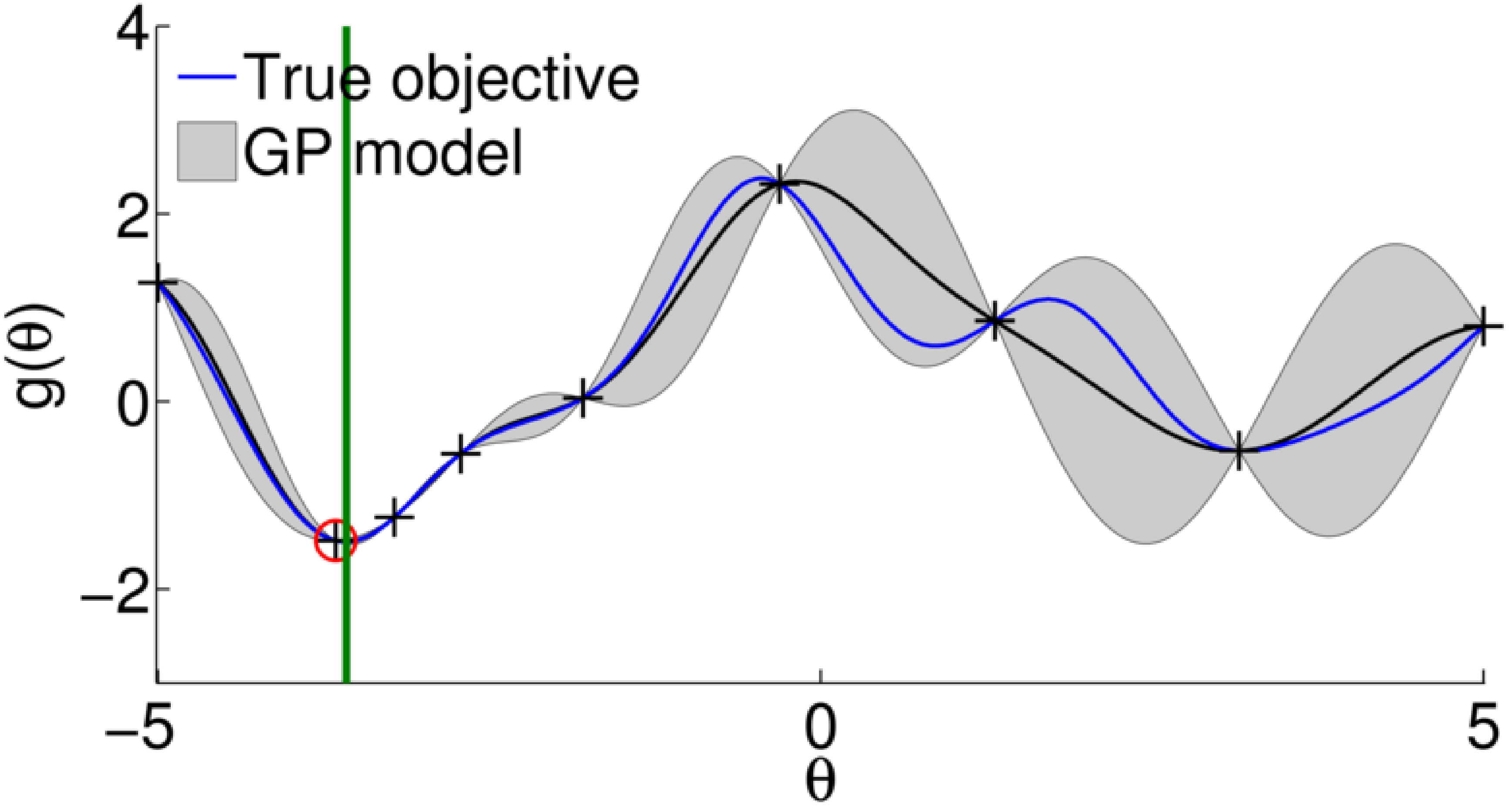
$$\boldsymbol{x}^* = \arg\min \tilde{f}(\boldsymbol{x})$$

Institución Universitaria
Reacreditada en Alta Calidad

Optimización Bayesiana

- Aprender la superficie de respuesta $\tilde{f}(\boldsymbol{x})$
- Basado en la superficie, seleccione los siguientes parámetros a evaluar $\boldsymbol{x}^{(t+1)}$
- Evaluar $\boldsymbol{x}^{(t+1)}$ de la función objetivo
- Repetir hasta que se cumpla un criterio de parada

Optimización Bayesiana



Contenido

1. Introducción.
2. Procesos Gaussianos – Superficie de respuesta.
3. Técnicas para NLPs.

Superficie de respuesta

Modelo sustituto (surrogate) que se requiere para aproximar la función objetivo a partir de los datos disponibles

$$D = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}, i = 1, \ldots, n \qquad \tilde{f}(x_i)\Big|_D \sim f(x)$$

Existen una gran cantidad de técnicas en la literatura:

- Funciones polinomiales.
- Random forests.
- Bayesian neural networks.
- Deep neural networks.
- Gaussian Processes. ← El más utilizado en la actualidad
- etc

Procesos Gaussianos

- Modelo de regresión flexible (no paramétrico)
- Distribución sobre funciones $\tilde{f} \sim GP(m_f, k_f)$
- Modelo probabilístico

$$y = \tilde{f}(X) + \epsilon\,, \quad \epsilon \sim N(0, \sigma^2)$$

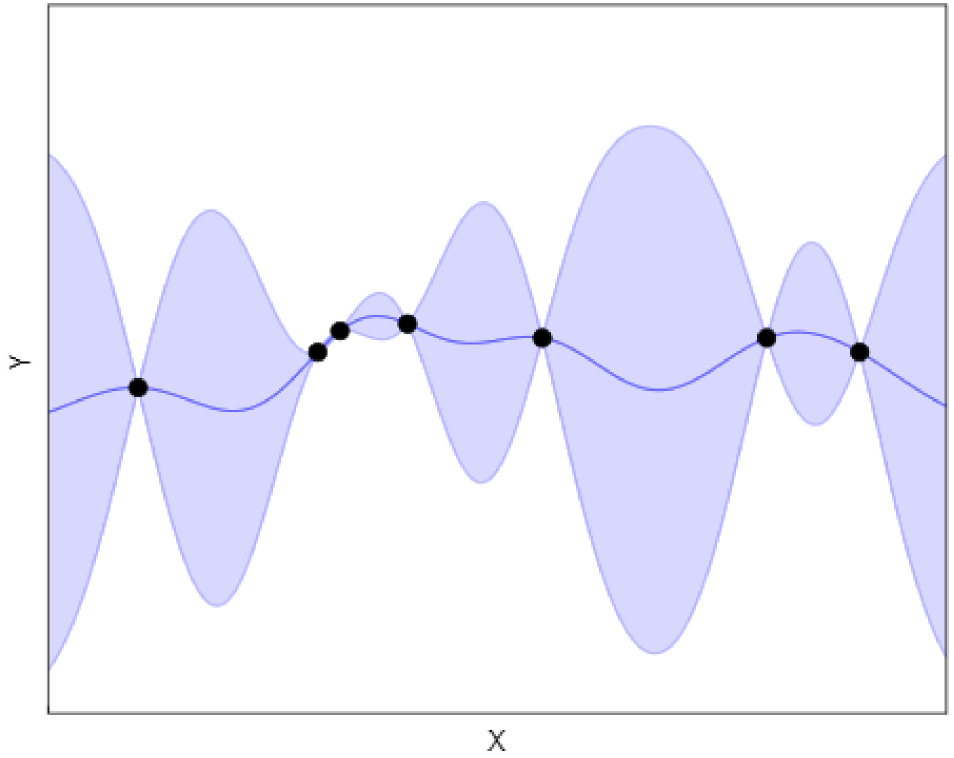
- El posterior de la distribución predictiva para una entrada arbitraria

$$p(\tilde{f}(x^*)|D, x^*) \sim N(\mu, \sigma^2)$$

$$\mu|_{x^*} = k(X, x^*)^T k(X, X)^{-1} y$$

$$\sigma^2|_{x^*} = k(x^*, x^*) - k(X, x^*)^T k(X, X)^{-1} k(X, x^*)$$

Procesos Gaussianos



Procesos Gaussianos

$$k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \delta_{ij}\sigma_n^2$$

Parámetros del GP
(hiperparametros)

Por que Procesos Gaussianos?

Pro:
- Matemática bien definida.
- Incertidumbres estimadas.
- Posibilidad de incluir información previa.
- Fácil forzar la suavidad.
- Buenas capacidades de modelamiento en regímenes de baja cantidad de datos.

Cons:
- Difícil escalar a entradas con alto dimensionalidad.
- Computacionalmente costoso.
- La calidad del modelo depende del uso apropiado del kernel.

Función de adquisición

Como seleccionamos los siguientes parámetros a evaluar $x^{(t+1)}$?

$$x^* = \arg\min \quad \alpha(\tilde{f}(x), D)$$

Intuición: Una buena función de adquisición $\alpha()$ necesita tener un balance entre exploración y explotación.

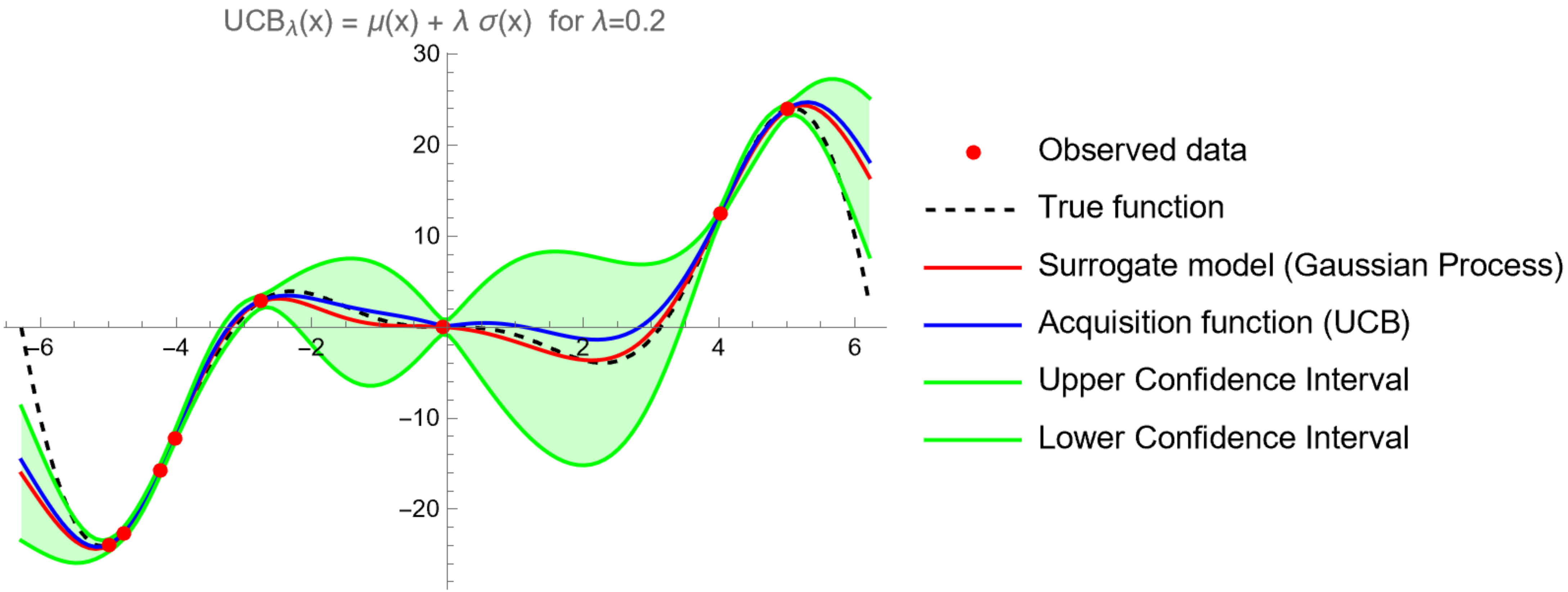
$$y = \alpha(\mu(x^*), \sigma(x^*))$$

Muchas funciones de adquisición en la literatura:

- Probabilidad de mejoramiento.
- Mejoramiento esperado.
- Limite de confianza superior.
- Búsqueda de entropía.
- Búsqueda de entropía predictiva

Limite de confianza superior

$$\mathrm{UCB}(x;\lambda) = \mu(x) - \lambda\sigma(x)$$

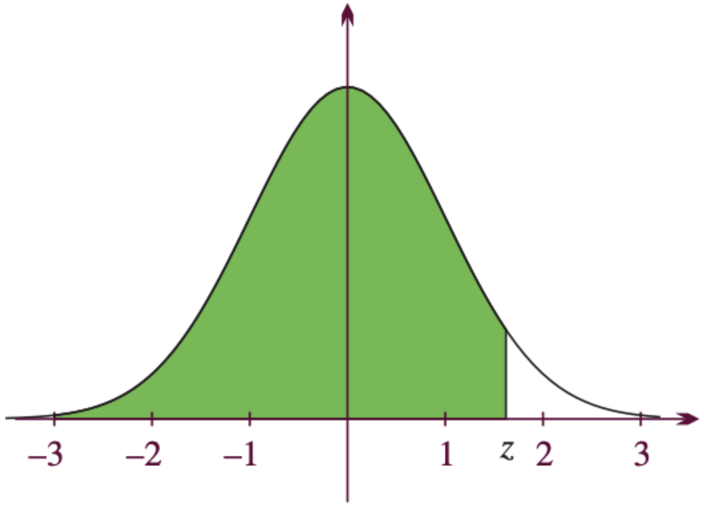


https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html

Expected Improvement

$$\mathrm{EI}(x;\xi) = (\mu - \tilde{f}(x) - \xi)\Phi\left(\frac{\mu - \tilde{f}(x) - \xi}{\sigma}\right) + \sigma\phi\left(\frac{\mu - \tilde{f}(x) - \xi}{\sigma}\right)$$

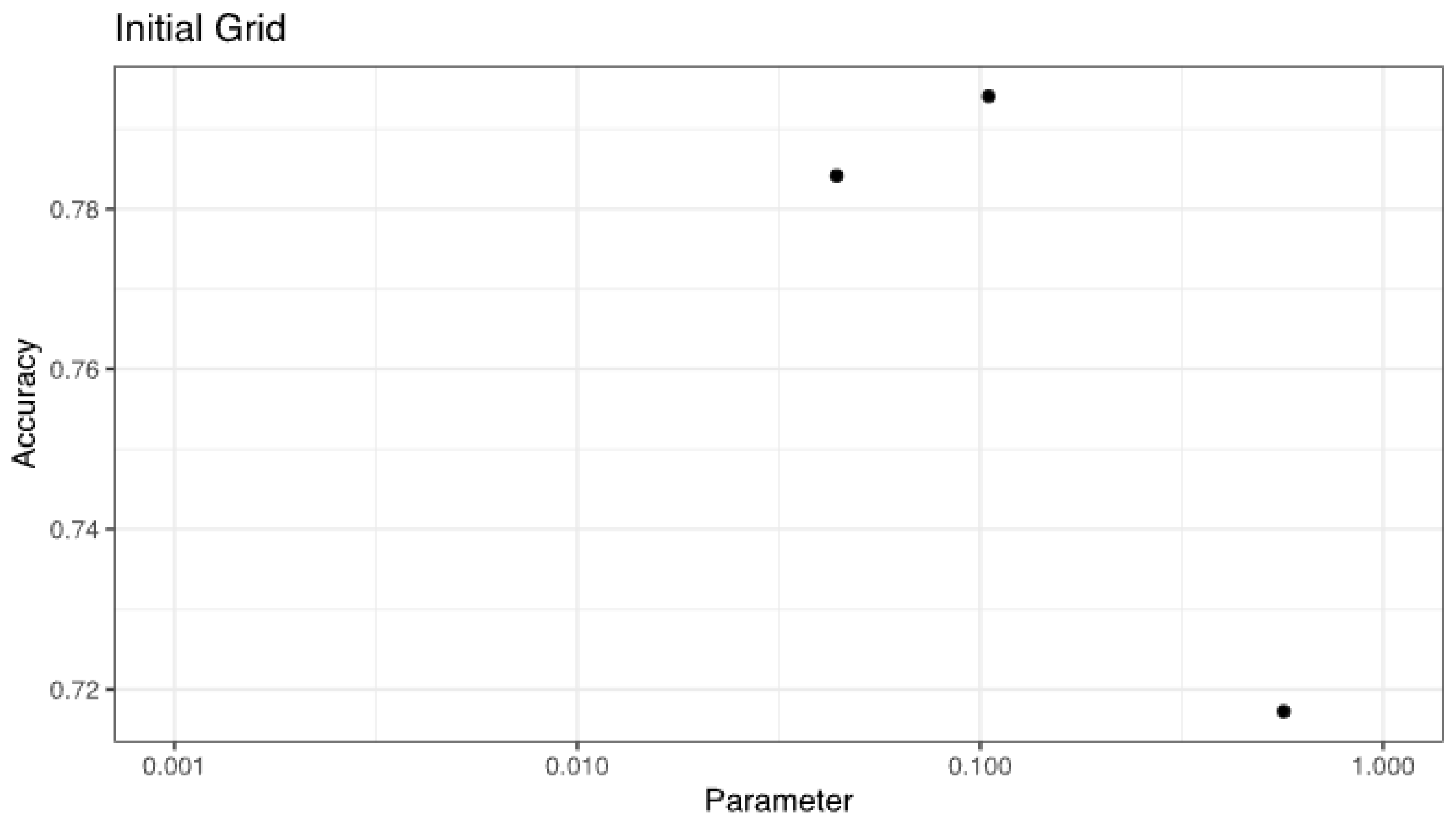
$$\Phi(z) = \int_{-\infty}^{z} \phi(t)\, dt.$$



ξ igual a 0 realiza explotación.
ξ grande realiza exploración.

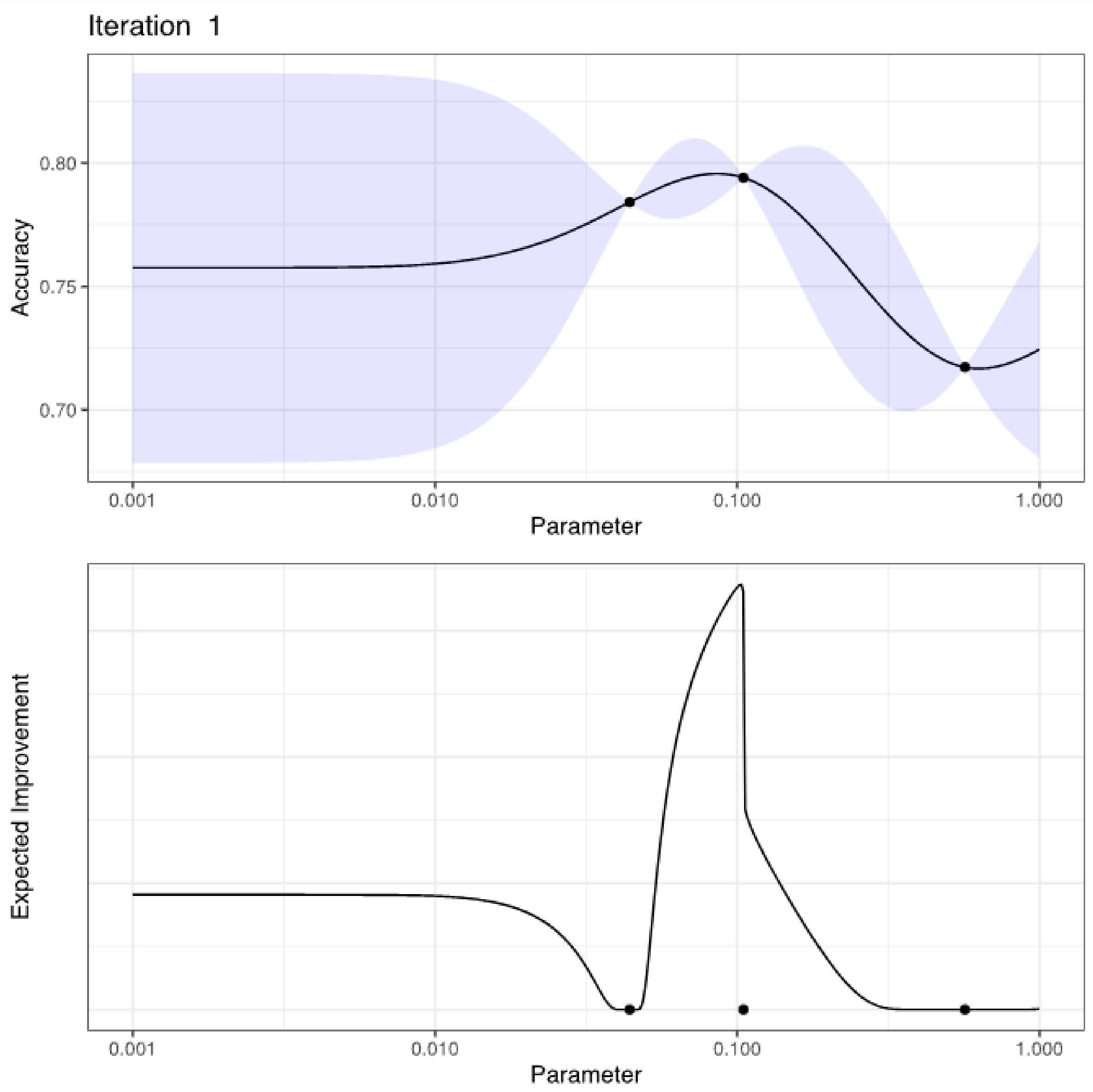
https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html

Expected Improvement

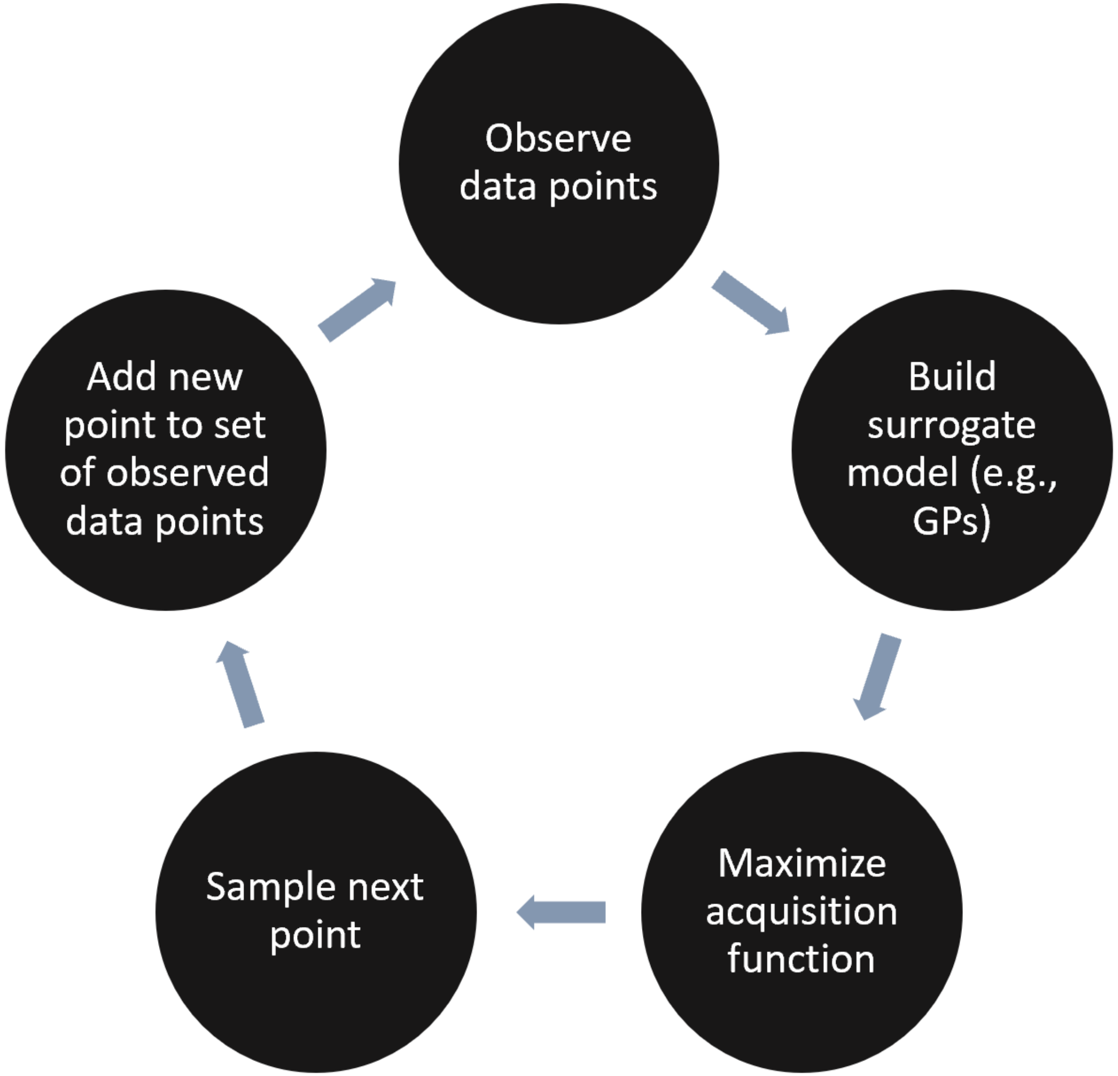


Institución Universitaria
Reacreditada en Alta Calidad

Expected Improvement



Algoritmo


Observe data points
Build surrogate model (e.g., GPs)
Maximize acquisition function
Sample next point
Add new point to set of observed data points

Algoritmo

Algorithm 1: Single-objective Bayesian optimization

$\mathcal{D} \longleftarrow$ if available: $\{\boldsymbol{x}, f(\boldsymbol{x})\}$
Prior $\longleftarrow$ if available: Prior of the response surface
repeat
    Train response surface $\hat{f}$ from $\mathcal{D}$
    Find $\boldsymbol{x}^*$ maximizing the acquisition surface $\alpha(\boldsymbol{x})$
    Evaluate $\boldsymbol{x}^*$ on the real system
    Add $\{\boldsymbol{x}^*, f(\boldsymbol{x}^*)\}$ to $\mathcal{D}$
until *stopCriteria*

Limitaciones de la optimización Bayesiana

Cons:

- No escala bien para espacio de parámetros de alta dimensionalidad (~30 dimensiones).
- No garantiza la convergencia.
- Si la función original es difícil de modelar, se obtienen desempeños muy malos (funciones disconstinuas, o con muchos minimos locales).

Pro:

- Sorpresivamente eficiente es muchas aplicaciones.
- Fácilmente se pueden incluir estructura o evaluaciones pasadas para un problema en particular.
- Modelos interpretables pueden proveer información.
- Funciona en el mundo real!

Referencias

- Basada en la presentación “Introduction to Bayesian Optimization” por el profesor Roberto Calandra. https://slides.com/rcalandra/introduction-to-bayesian-optimization

Institución
Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín