Advanced Hyperparameter Optimization

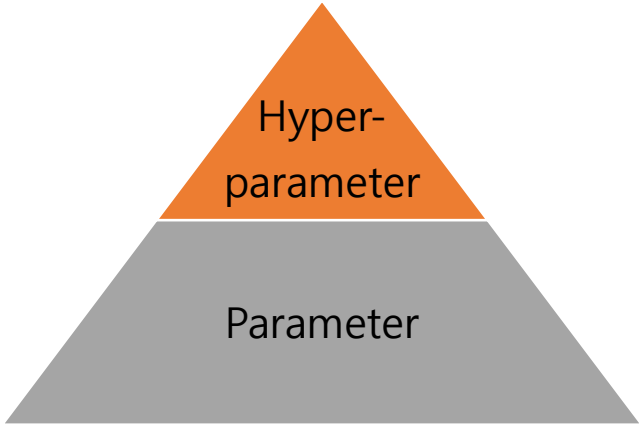
2020-10-16

손형욱

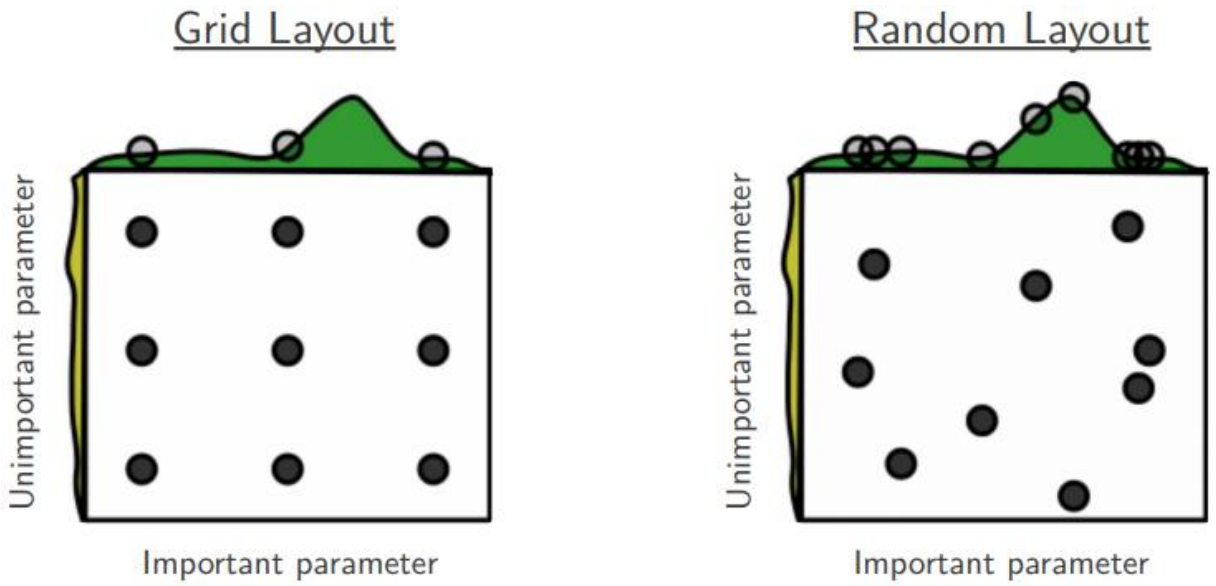
Parameters and Hyperparameters

하이퍼파라미터 최적화 문제

- 파라미터와 하이퍼파라미터:
  - θ = (weight, BN scale factor, …)
  - η = (레이어 수, learning rate, weight decay, …)
- Optimization 과 Meta-optimization
  - $\mathcal{L}^* = \min_{\eta} \min_{\theta} \mathcal{L}(x, y; \theta, \eta)$
- 왜 하이퍼파라미터 최적화가 어려운가?
  - $\mathcal{L}(\eta) = \min_{\theta} \mathbb{E}_{\mathcal{D}}[\mathcal{L}(x, y; \theta, \eta)]$ 의 계산이 매우 expensive 함.
  - No access to gradient $\nabla_{\eta} \mathcal{L}(\eta)$.



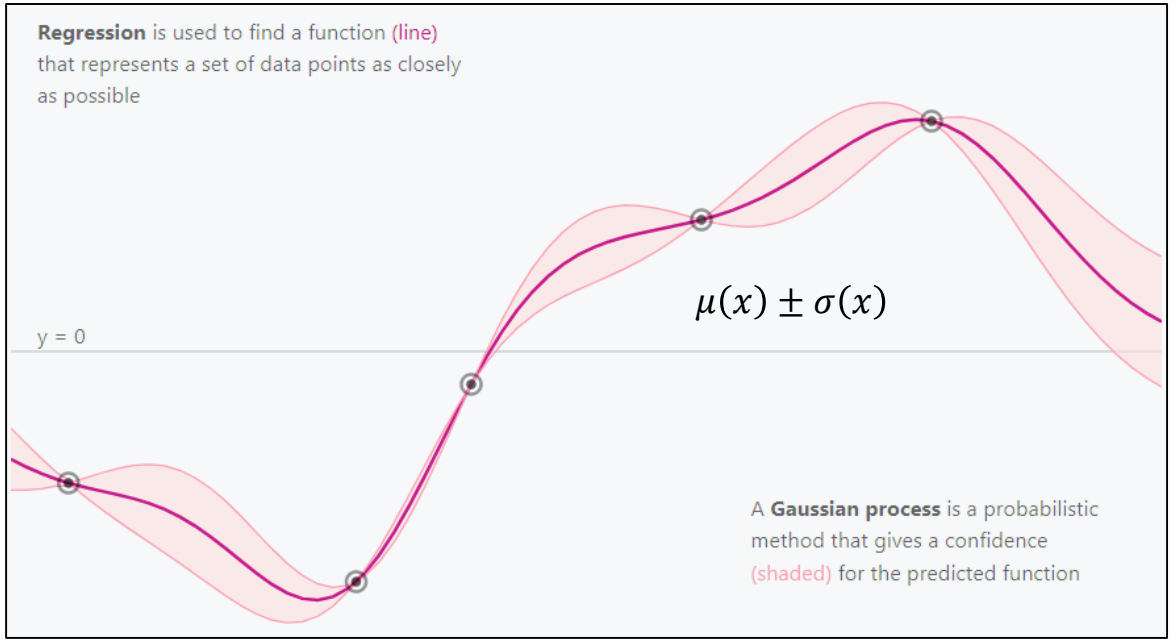
Typical methods (grid, random)



어떻게 하이퍼파라미터 공간을 더 효율적으로 탐색할 수 있을까?
→ 관심없는 영역은 적게 탐색하고, 관심있는 영역은 많이 탐색하자!
→ = Bayesian Optimization (BO)

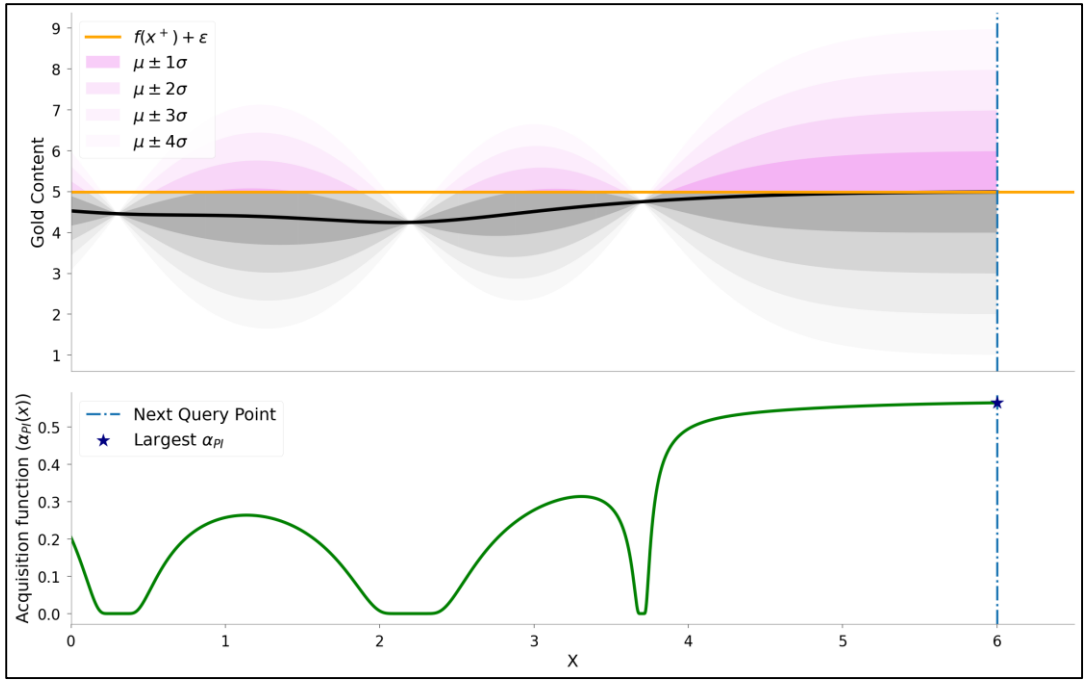
베이지안 최적화의 핵심 요소들

Gaussian Process (GP) regression



GP 는 관측된 값으로 함수의 분포를 모델링한다.

Acquisition function



Acquisition function 을 이용해 query point 를 결정한다.

Gaussian process regression

Stochastic processes

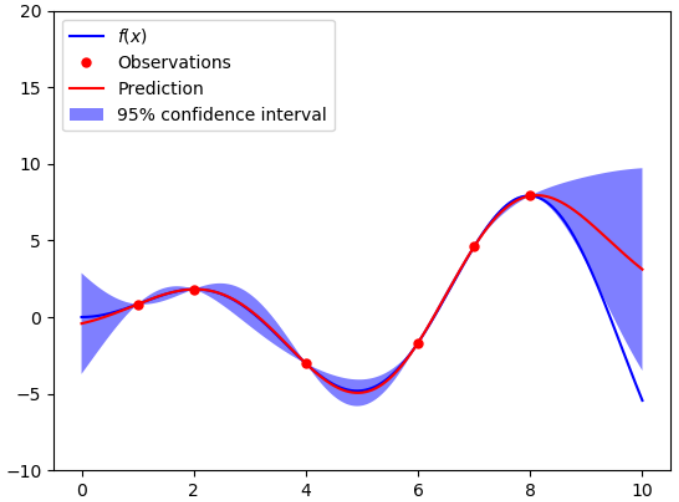
- 시계열 데이터를 모델링하는 방법.
  - 관측치 $\mathcal{D} = \{x(t)\}_1^T$ 가 있을 때 결측치를 예측하는 방법? (interpolation 문제)
- Deterministic methods
  - linear/quadratic regression, …
  - MSE 가 최소인 함수를 선택. (예측값이 deterministic 함)
- Stochastic processes
  - 시퀀스를 확률변수의 집합으로 본다. $\boldsymbol{X}[t] = (X_1, X_2, \dots, X_n)$
  - 확률변수들의 결합분포 $p(X_1, X_2, \dots, X_n)$ 를 모델링하자. (= 함수 $\boldsymbol{X}[t]$ 의 분포를 학습)

GP ∈ Stochastic Processes

https://distill.pub/2019/visual-exploration-gaussian-processes/

- Gaussian Process
  - $p(X_1, X_2, \ldots, X_n)$ 를 n차원 가우시안 분포 $\mathcal{N}(\mu, \Sigma)$ 로 정의함.
  - 목표는 (μ, Σ) 를 학습하는 것.
  - 시간축 t 가 연속이면 $\mathrm{n} \to \infty$ 인 케이스에 해당.
- 학습은 어떻게 하나?
  - 관측치 (t, x_t) 가 주어졌을 때,
  - 조건부 확률분포 $p(X_1, X_2, \ldots, X_n | X_t = x_t)$ 를 계산하는 것과 같음.
  - 베이즈 정리를 이용해 조건부 확률분포 계산.

연속시간 GP



$$\boldsymbol{X}(t) \sim \mathcal{N}\big(\mu(t), \Sigma(t, t')\big)$$

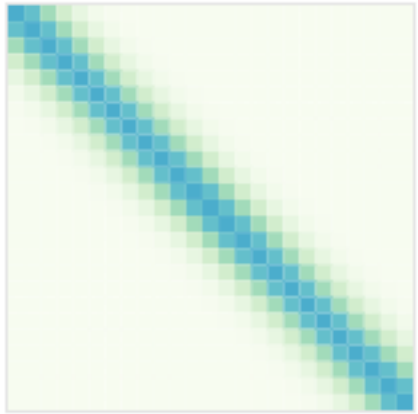
Gaussian Process

- GP에서 결합분포의 공분산 행렬 Σ 을 kernel 이라고 부름.
- Kernel 은 함수 κ: $(t, t') \to \mathbb{R}$ 로 정의됨. (예시 →)
- 즉, $\kappa(t, t') = \Sigma_{t,t'} = \mathrm{Cov}(X_t, X_{t'})$ 을 표현함.
- 왜 가우시안 분포를 쓰나?
  - 계산이 간단해지기 때문
  - Conditional distribution 과 marginal distribution 도 가우시안 분포.
  - 결과도 closed-form 으로 나옴.

대표적인 커널의 예시

RBF KERNEL

$$\sigma^2 \exp\left(-\frac{||t-t'||^2}{2l^2}\right)$$



Variance σ = 0.8

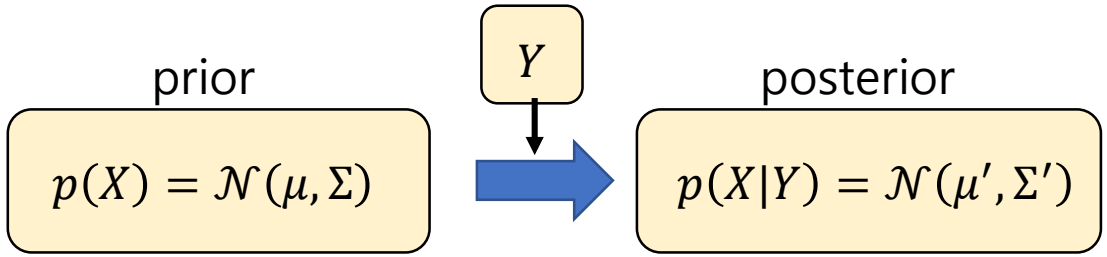
Length l = 0.8

Update rule for GP

- 조건부 확률 계산=Bayesian inference (AKA, Bayesian update rule)

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)}$$

- 관측치 $\boldsymbol{Y}$ 를 사용해 사전 확률분포 $p(\boldsymbol{X})$ 로부터 사후 확률분포 $p(\boldsymbol{X}|\boldsymbol{Y})$ 를 구한다.



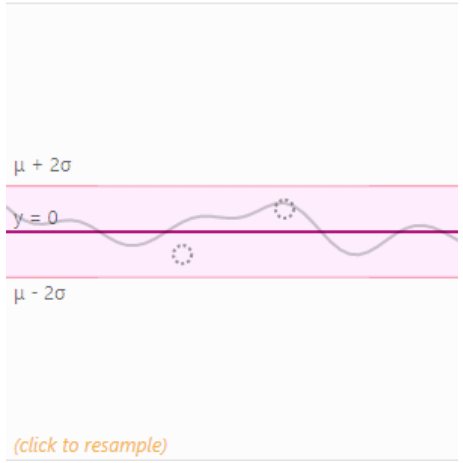
- 가우시안의 조건부 확률분포는 닫힌 해가 계산됨.

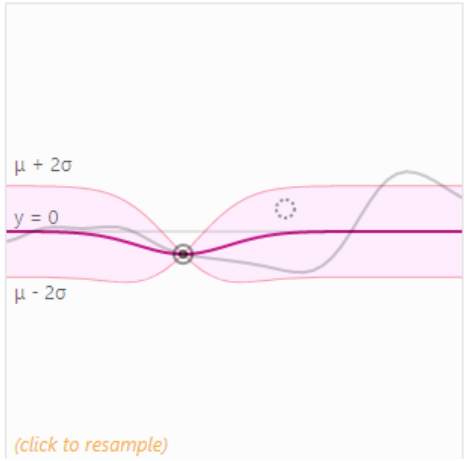
$$\mu' = \mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y)$$

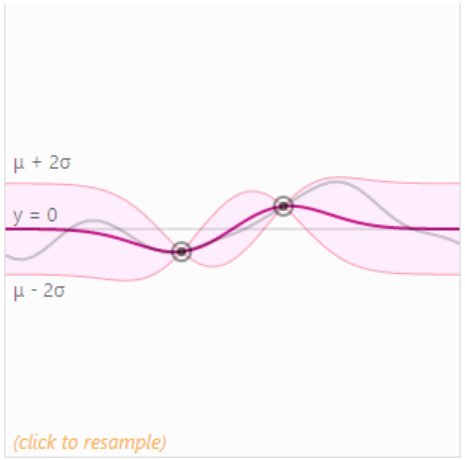
$$\Sigma' = \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}$$

GP after each observations

value distribution (marginal distribution)



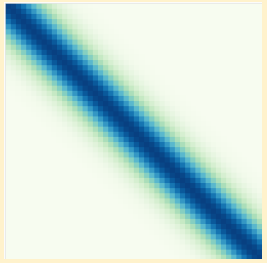




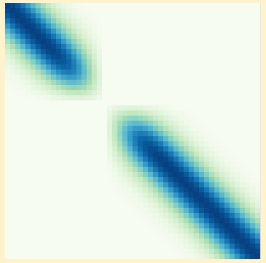
prior

$p(\boldsymbol{X})$:

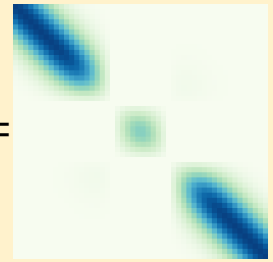
$p(\boldsymbol{X}) = \mathcal{N}(\mu, \Sigma)$

$\Sigma =$ 

$p(\boldsymbol{X}|\mathcal{D}_1) = \mathcal{N}(\mu', \Sigma')$

$\Sigma' =$ 

$p(\boldsymbol{X}|\mathcal{D}_1, \mathcal{D}_2) = \mathcal{N}(\mu'', \Sigma'')$

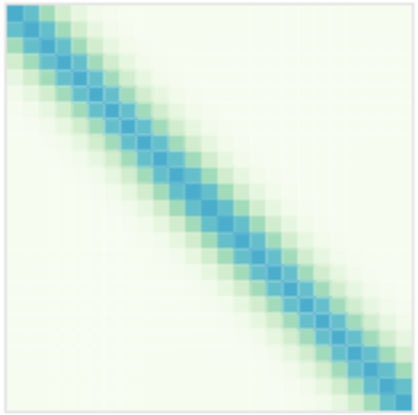
$\Sigma'' =$ 

여러 종류의 prior kernel

- Prior 의 kernel 은 예측모형의 성질을 결정함. (locality, periodicity, linearity, …)
- 여러 종류를 선형 결합해 사용할 수 있음.

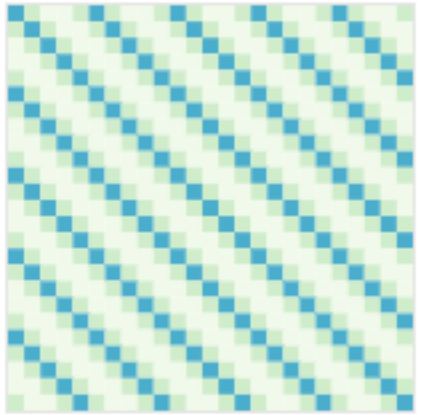
RBF KERNEL

$$\sigma^2 \exp\left(-\frac{||t-t'||^2}{2l^2}\right)$$



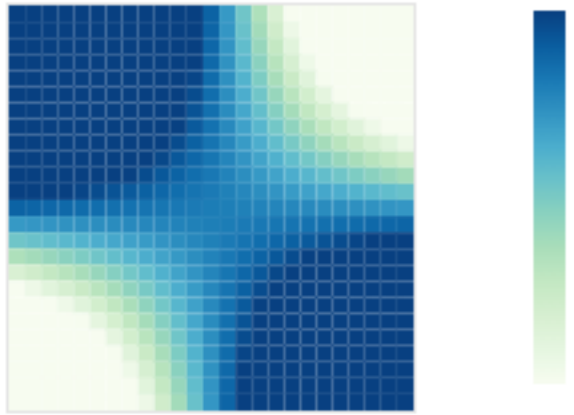
PERIODIC

$$\sigma^2 \exp\left(-\frac{2\sin^2(\pi|t-t'|/p)}{l^2}\right)$$



LINEAR

$$\sigma_b^2 + \sigma^2(t-c)(t'-c)$$

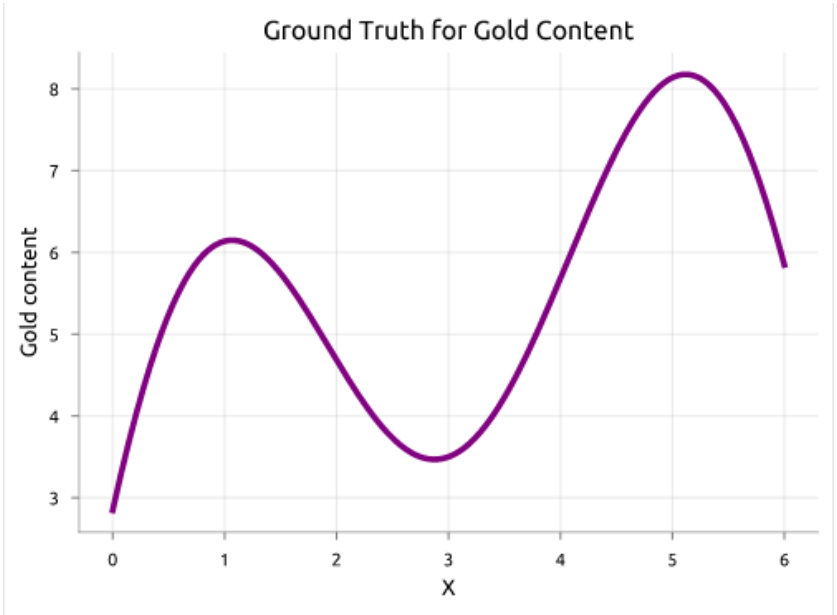


Bayesian Optimization

베이지안 최적화

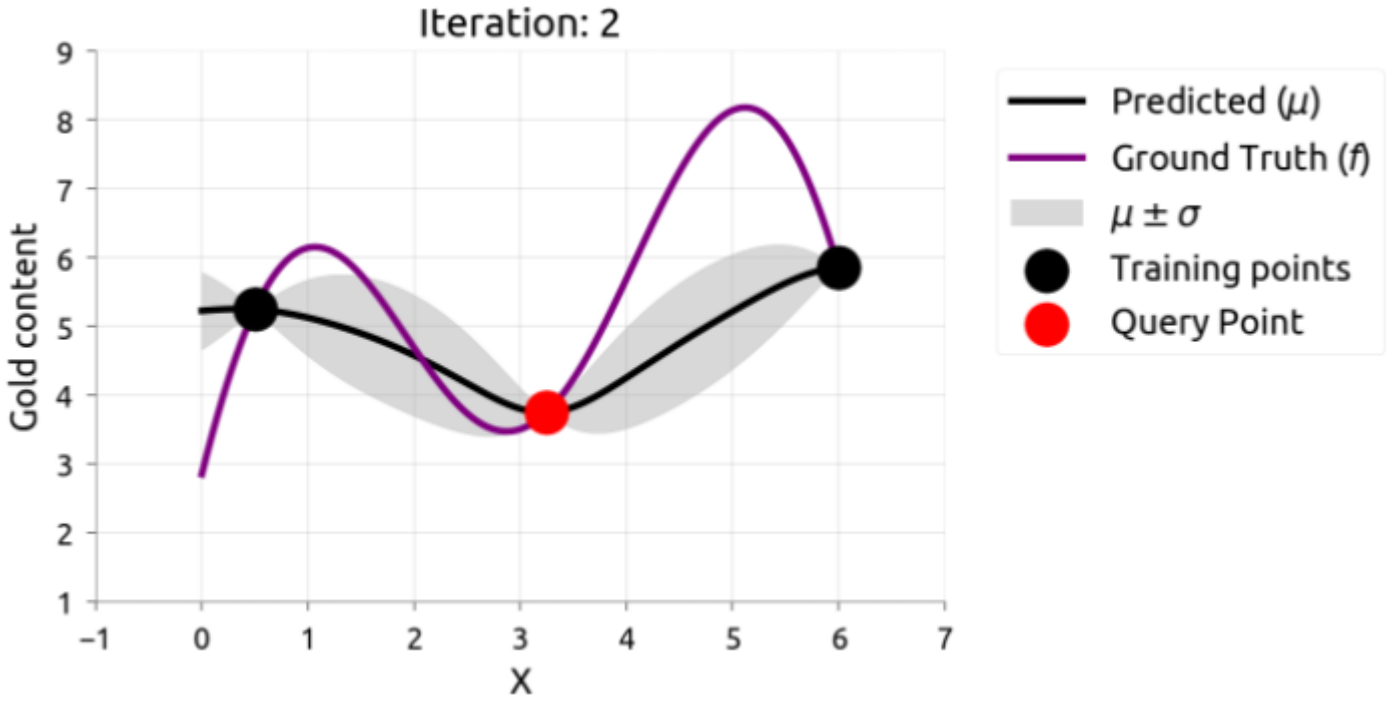
- 몇 번의 실험결과 $\mathcal{D} = \{f(x_i)\}_1^t$ 가 주어졌을 때, query point x_{t+1} 을 고르는 방법?

1. 랜덤하게 t 개의 지점 $\{x_1, x_2, \dots, x_t\}$ 을 선택해 $\mathcal{D} = \{f(x_i)\}_1^t$ 를 계산한다.
2. $f(x)$ 를 예측하는 surrogate model 을 학습한다. (GP 사용)
3. 성능이 높을 것으로 예측되는 지점 x_{t+1} 을 query point 로 선택한다.
4. 관측된 최대값 $\max\{f(x_i)\}$ 가 수렴할 때까지 반복.



Surrogate model

- GP 를 이용해 관측치 $\{f(x_1), f(x_2), \ldots\}$ 로부터 $f(x)$ 을 예측하는 모델을 두자.
- 각 지점 x 에서 기대값 $\mu(x)$ 과 uncertainty $\sigma(x)$ 를 예측함.



3개 지점을 관측했을 때 GP

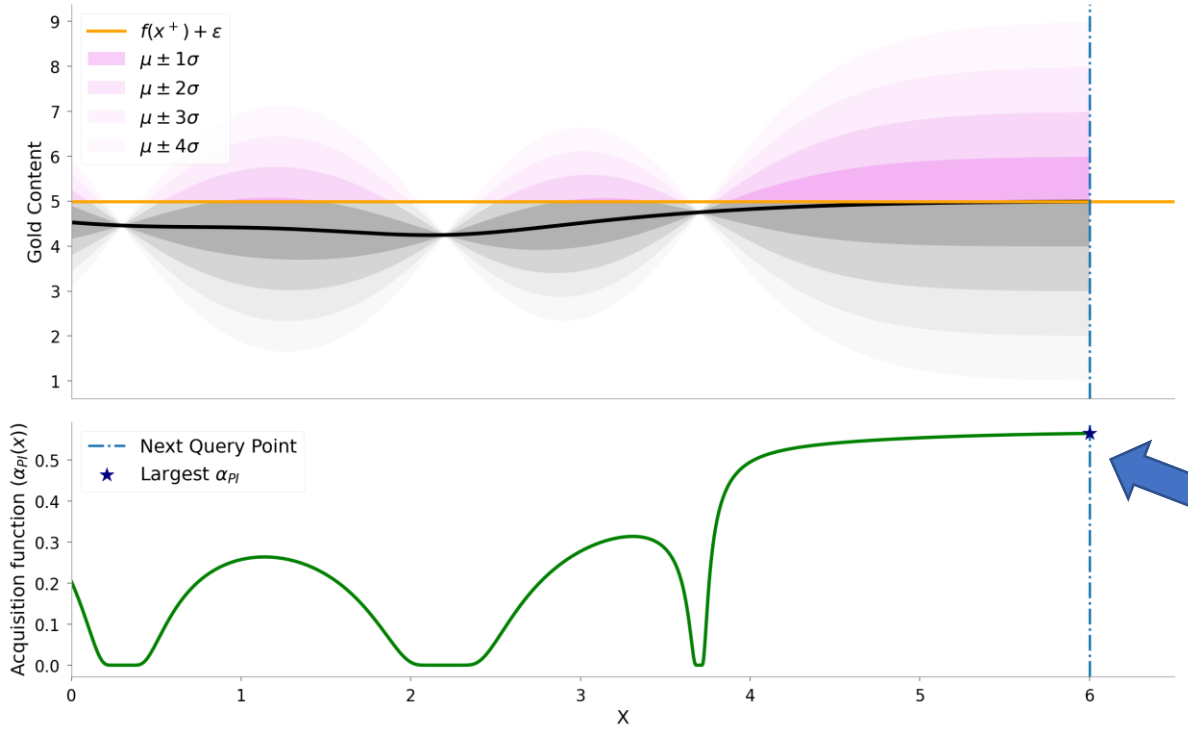
https://distill.pub/2020/bayesian-optimization/

Acquisition function

- 현재까지의 샘플을 이용해 다음 query point (x_{t+1}) 을 선택하는 방법?
- GP 모델의 아웃풋이 확률분포임을 이용하자.
- 예를 들어, $\Pr[f(x) > f(x^+)]$ 가 최대가 되는 지점을 query point 로 선택!

$p(f(x))$

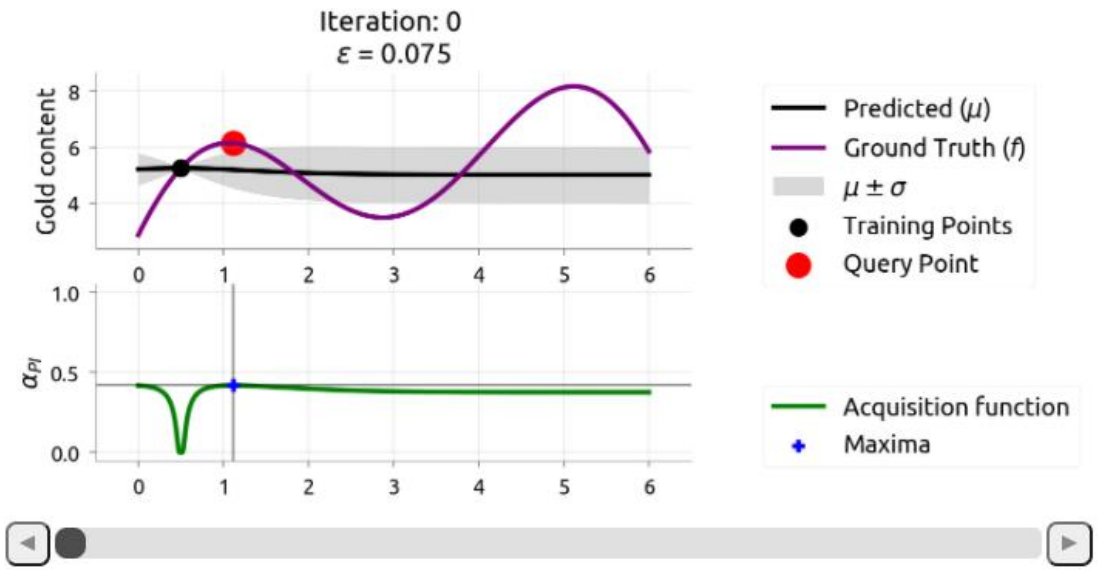
$\alpha(x)$

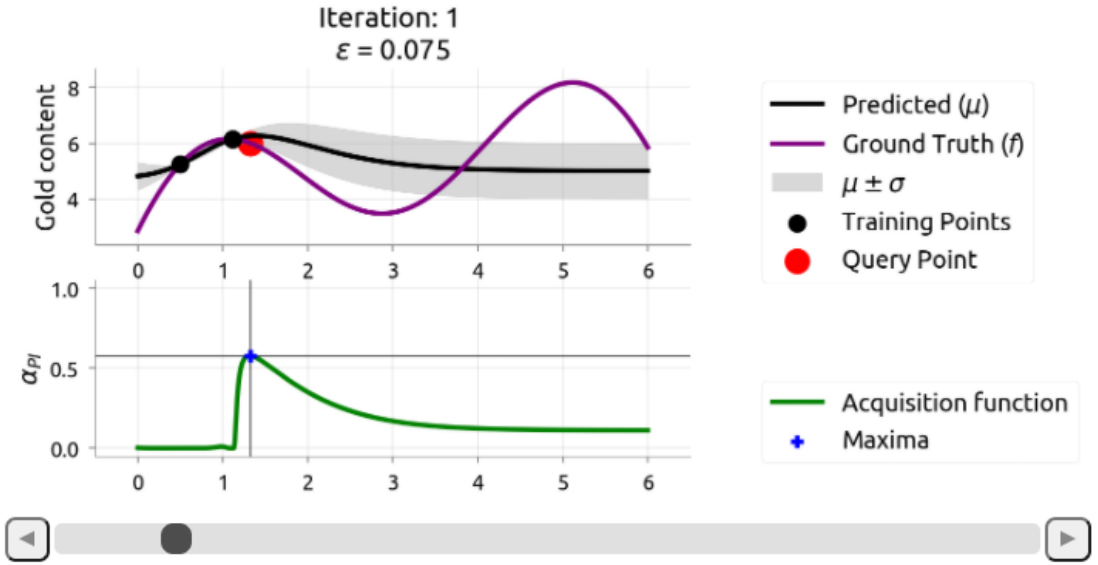


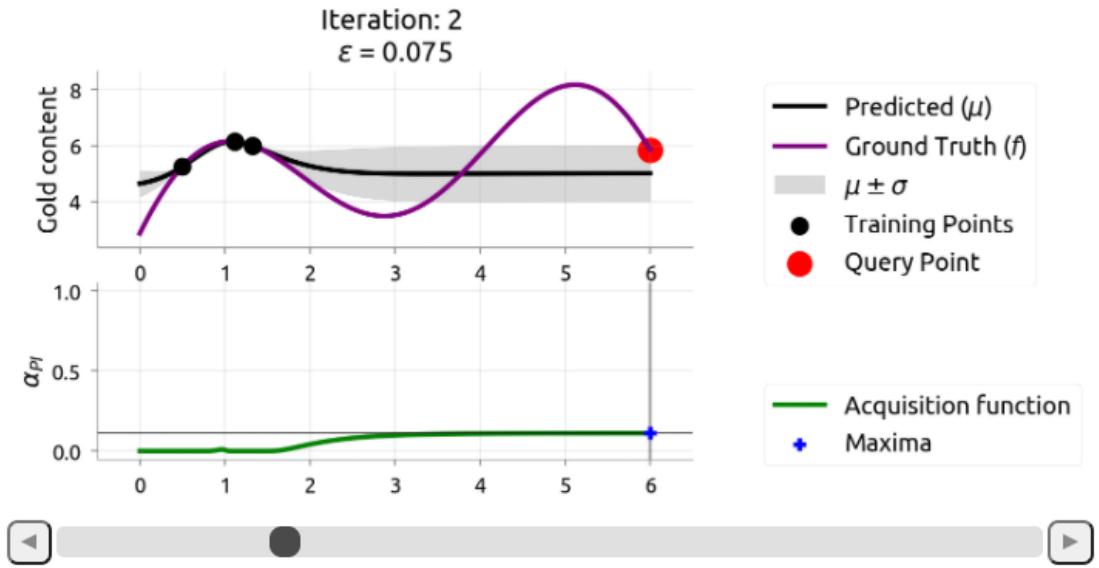
x^+: current maxima

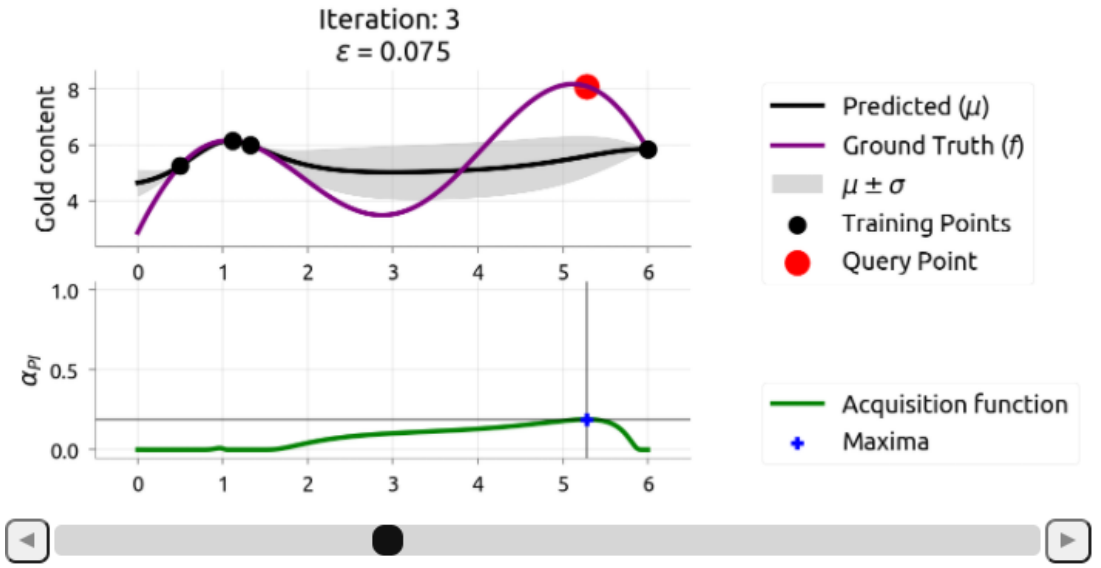
Next query point

$x_{t+1} = argmax_x\, \alpha(x)$


Iteration: 0
ϵ = 0.075
Gold content
α_{PI}
Predicted (μ)
Ground Truth (f)
$\mu \pm \sigma$
Training Points
Query Point
Acquisition function
Maxima


Iteration: 1
ϵ = 0.075
Gold content
α_{PI}
Predicted (μ)
Ground Truth (f)
$\mu \pm \sigma$
Training Points
Query Point
Acquisition function
Maxima


Iteration: 2
ϵ = 0.075
Gold content
α_{PI}
Predicted (μ)
Ground Truth (f)
$\mu \pm \sigma$
Training Points
Query Point
Acquisition function
Maxima


Iteration: 3
ϵ = 0.075
Gold content
α_{PI}
Predicted (μ)
Ground Truth (f)
$\mu \pm \sigma$
Training Points
Query Point
Acquisition function
Maxima

다양한 acquisition function 들

Type	Definition
Probability of Improvement (PI)	$\alpha_{\mathrm{PI}}(x) = \Pr[f(x) > f(x^+) + \epsilon]$
Expected Improvement (EI)	$\alpha_{\mathrm{EI}}(x) = \mathbb{E}[\max\{0, f(x) - f(x^+) + \epsilon\}]$
Upper Confidence Bound (UCB)	$\alpha_{\mathrm{UCB}}(x) = \mu(x) + \lambda \times \sigma(x)$
EI 와 PI 의 선형 결합	$\alpha_{\mathrm{EI-PI}}(x) = \alpha_{\mathrm{EI}}(x) + \lambda\alpha_{\mathrm{PI}}(x)$

$$x_{t+1} = argmax_x\, \alpha(x)$$

x^+: current maxima

ϵ: exploration-exploitation rate

A step back: what does BO really do?

- $\max_x f(x)$ 문제를 $\max_x \alpha(x)$ 문제로 대체한 것임.
- 즉 여전히 최적화 문제를 풀어야 함. $\alpha(x)$ 가 convex 하다는 보장도 없음.
- 그럼 왜 원래 문제 대신 $\alpha(x)$ 최적화 문제를 풀까?
- $\rightarrow$ $\alpha(x)$ 와 $\nabla\alpha(x)$ 의 계산이 훨씬 쉽기 때문.

Hands-on

- A number of optimization tools implement BO.
- Scikit-optimize
  - skopt.gp_minimize()
  - You define the objective function and search space.

https://colab.research.google.com/drive/1sRgVZtvRt_kS84nHi5JhNAQQOP5me6wS?usp=sharing

Tips

- If your search space has huge dynamic range, perform search in the log-space: $\eta' = \log \eta$ (ex. Learning rate, regularization term)
- BO is designed to minimize the number of evaluations. Also, GP inference is $\mathcal{O}(N^2)$ to number of input points.

BO 가 유용한 경우

- 성능이 하이퍼파라미터에 sensitive 할 때.
- 바뀐 조건 (데이터셋, 네트워크) 에 맞추어 하이퍼파라미터를 fine-tuning 해야 할 때.
- 새로 정의한 하이퍼파라미터의 적합한 range 를 모를 때.

https://distil.pub