Bayesian Optimization

Hongwei Jin

November 16, 2018

Department of Computer Science
U. of I. at Chicago

Outline

Overview

Big Picture

Optimization

A function f is smooth, we can apply

- first-order method: gradient descent, SGD, etc.
- second-order method: Newton's method, L-BFGS, etc.

What if the function has no first- and second-order information?

Black-box optimization

Problem

Problems

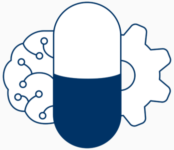
To find the "global minimizer" of $f(\boldsymbol{x}) \to \mathbb{R}$ where $\boldsymbol{x} \subseteq \mathbb{R}^d$ is a "bounded domain":

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmax}} f(\boldsymbol{x})$$

- f is explicitly unknown function without first- and second-order information
- f is expensive to evaluate, but $f(\boldsymbol{x})$ is accessible for all $\boldsymbol{x} \in \mathcal{X}$
- f is Lipschitz-continuos, i.e., $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq c \|\boldsymbol{x} - \boldsymbol{x}'\|$

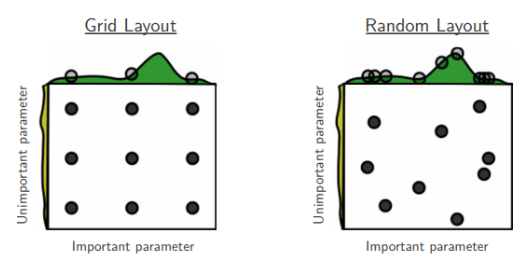
Application

- tunning hyperparameters: number of layers/number of units per layer, learning rate, regularizer, etc.
- designing experiments: physical, chemistry, biological experiments,
- expensive evaluations: drug trial (time consuming), financial investments (money consuming).



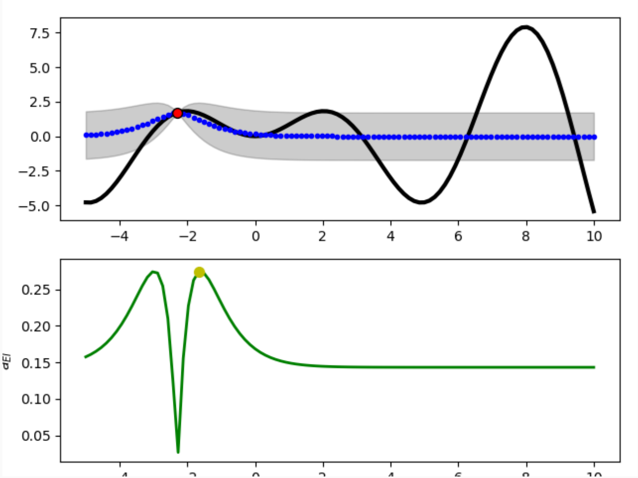
Approaches

- Experience: assign hyperparameters based on expert knowledges
- Grid search: search a hypercube of the hyperparameters
- Random search: sample the hypercubc uniformly, better than grid search, but still expensive [1]



- Bayesian optimization (BO): search the domain based on the Gaussian processes

[1] Bergstra and Bengio, JMLR 2012


7.5
5.0
2.5
0.0
−2.5
−5.0
−4
−2
0
2
4
6
8
10
0.25
0.20
0.15
0.10
0.05

Pre-knowledge

Gaussian Process (GP)

Optimizing over the function $\Rightarrow$ predict the function based on "small set" of data

Consider the problem of nonlinear regression: you want to learn a function f from data $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$

Gaussian process can be interpreted as a prior over function:

$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

GP cont'd

Definition (GP)

A **gaussian process** is a collection of random variables, and finite number of which have a joint Gaussian distribution.

GP is determined by

- mean function:

$$m(\boldsymbol{x}) = \mathbb{E}\left[f(\boldsymbol{x})\right]$$

- covariance function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))\right]$$

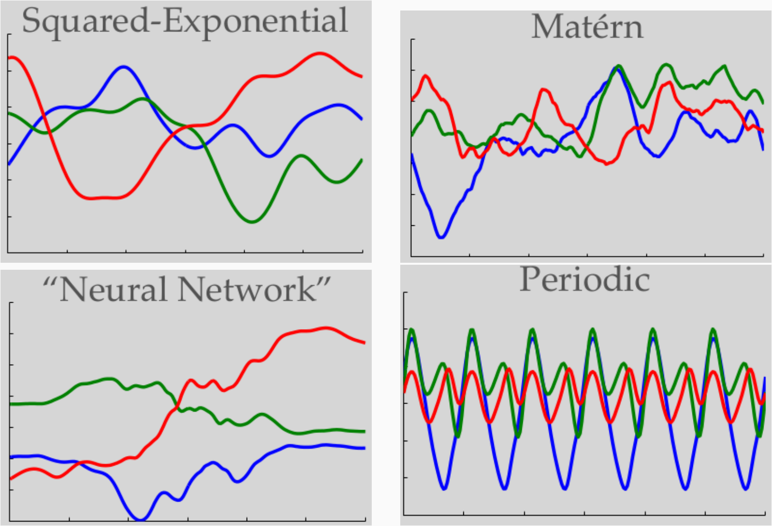
Denoted as:

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$$

Examples of Covariance Functions

- squared exponential: $k_{SE}(r) = \exp\left(-\frac{r^2}{2\ell^2}\right)$
- Matérn class: $k_{Matern}(r) = \frac{2^{1-\mu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}r}{\ell}\right)$
- Exponential: $k_{Exp}(r) = \exp\left(-\frac{r}{\ell}\right)$
- γ-exponential: $k_{\gamma-Exp}(r) = \exp\left(-\left(\frac{r}{\ell}\right)^{\gamma}\right)$, for $0 < \gamma \leq 2$
- Neural network: $k_{NN}(\boldsymbol{x}, \boldsymbol{x}') = \frac{2}{\pi}\sin^{-1}\left\{\frac{2\boldsymbol{x}^{\top}\Sigma\boldsymbol{x}'}{\sqrt{1+2\boldsymbol{x}^{\top}\Sigma\boldsymbol{x}}\sqrt{1+2\boldsymbol{x}'^{\top}\Sigma\boldsymbol{x}'}}\right\}$
- Periodic: $k_{periodic}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\frac{2\sin^2\left(\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')\right)}{\ell^2}\right\}$

Example of Covariance Function cont'd



Covariance Functions

A stationary covariance function is a function of $\boldsymbol{x} - \boldsymbol{x}'$

A covariance function is called isotropic if it is a function only of $\|\boldsymbol{x} - \boldsymbol{x}'\|$.

$$r = \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|$$

is also called radial basis functions (RBFs)

Example (SE covariance function)

$$k_{SE}(r) = \exp\left(-\frac{r^2}{2\ell^2}\right)$$

Covariance Functions cont'd

A covariance function is called dot product covariance function, if it is a function depends only on $\boldsymbol{x}$ and $\boldsymbol{x}'$ through $\boldsymbol{x} \cdot \boldsymbol{x}'$

Example

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 + \boldsymbol{x} \cdot \boldsymbol{x}'$$
$$k(\boldsymbol{x}, \boldsymbol{x}') = (\sigma_o^2 + \boldsymbol{x} \cdot \boldsymbol{x}')^p, \quad p \in \boldsymbol{I}^+.$$

A more general name of the covariance function of taking two inputs $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{x}' \in \mathcal{X}$ into $\mathbb{R}$ is called kernel.
A covariance matrix $\boldsymbol{K}$ is a Gram matrix of pairwise covariance functions of a given points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$, where $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Matérn Class of Covariance Function

$$k_{Matern}(r) = \frac{2^{1-\mu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{\ell} \right)$$

ν, ℓ are positive parameters, and $K_{\nu}(\cdot)$ is a modified Bessel function of second kind.

A Bessel function is the canonical solutions $y(x)$ of Bessel's differential equation:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - \alpha^2) y = 0$$

where α is an arbitrary complex number.

The modified Bessel functions of the first and second kind are defined as

$$I_{\nu}(x) = \sum_{m=0}^{\infty} \frac{1}{m! \Gamma(m + \nu + 1)} \left(\frac{x}{2} \right)^{2m+\nu}, \quad K_{\nu}(x) = \frac{\pi}{2} \frac{I_{-\nu}(x) - I_{\nu}(x)}{\sin \nu \pi},$$

where ν is a positive non-integer.

Matérn Kernels cont'd

$$k_{Matern}(r) = \frac{2^{1-\mu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{\ell} \right)$$

- $\nu \to \infty$, then $k_{Matern}(r) = k_{SE}(r)$
- if $\nu = p + 1/2, p \in \boldsymbol{I}$, it has a simple form:

$$k_{\nu=p+1/2}(r) = \exp\left(-\frac{\sqrt{2\nu} r}{\ell} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=1}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu} r}{\ell} \right)^{p-i}$$

- most commonly used $\nu = 3/2$ and $\nu = 5/2$:

$$k_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3} r}{\ell} \right) \exp\left(-\frac{\sqrt{3} r}{\ell} \right), k_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5} r}{\ell} + \frac{5r^2}{3\ell^2} \right) \exp\left(-\frac{\sqrt{5} r}{\ell} \right)$$

[1] Rasmussen, William, 2006

Other Kernels

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{\ell}r\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{\ell}r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\big(-(\frac{r}{\ell})^{\gamma}\big)$	$\surd$	$\surd$
rational quadratic	$(1+\frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

S: stationary, ND: non-degenerate

Eigenfunction Analysis of Kernels

Idea: measure the nearness or similarity between data points

GPR can be viewed as Bayesian linear regression with a possibly infinite number of basis function

One of such basis function is eigenfunctions of the covariance functions.

Definition (eigenfunction)

A function $\phi(\cdot)$ that obeys the integral equation

$$\int k(\boldsymbol{x}, \boldsymbol{x}')\phi(\boldsymbol{x})d\mu(\boldsymbol{x}) = \lambda\phi(\boldsymbol{x}')$$

is called eigenfunction of kernel k with eigenvalue λ with respect to measure μ.

Kernels cont'd

Theorem (Mercer's Theorem)

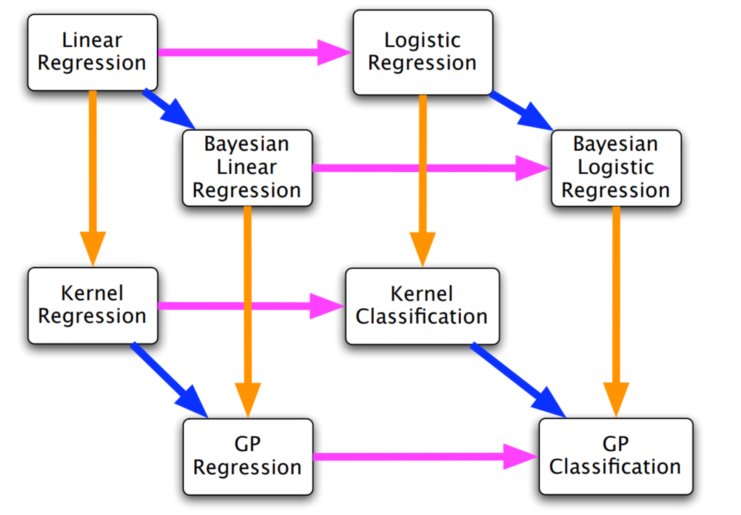
Let $(\mathcal{X}, \mu)$ be a finite measure space and $k \in L_\infty(\mathcal{X}^2, \mu^2)$ be a kernel such that $T_k : L_2(\mathcal{X}, \mu) \to L_2(\mathcal{X}, \mu)$ is positive definite. Let $\phi_i \in L_2(\mathcal{X}, \mu)$ be the normalized eigenfunctions of T_k associated with the eigenvalues $\lambda_i > 0$. Then:

1. *the eigenvalues $\{\lambda_i\}_{i=1}^\infty$ are absolutely summable*
2. $$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^\infty \lambda_i \phi_i(\boldsymbol{x}) \phi_i^*(\boldsymbol{x}'),$$

   holds μ^2 almost everywhere, where the series converges absolutely and uniformly μ^2 almost everywhere.

$\Rightarrow$ RKHS (Mercer's theorem, eigenfunction AND reproducing kernel map)

GP Summary



[1]Zoubin Ghahramani, NIPS 2016

GP Summary cont'd

- GPs define distributions on functions
- GPs are closely related to many other models:
  - Bayesian kernel machine
  - linear regression with basis functions
  - Deep neural networks
- GPs handle uncertainty in unknown function f by averaging

Now, if we want to minimize the unknown function, using GP as the prior of function
$\Rightarrow$ Bayesian optimization

Bayesian Optimization

BO in a Nutshell

Assume: $f(\boldsymbol{x}) \sim GP(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$

posterior: the point to query next

acquisition function: a utility function

Algorithm 1 Bayesian Optimization

1: **for** $t = 1, 2, \ldots$ **do**
2: Find $\mathbf{x}_t$ by optimizing the acquisition function over the GP: $\mathbf{x}_t = \mathrm{argmax}_{\mathbf{x}}\, u(\mathbf{x}|\mathcal{D}_{1:t-1})$.
3: Sample the objective function: $y_t = f(\mathbf{x}_t) + \varepsilon_t$.
4: Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (\mathbf{x}_t, y_t)\}$ and update the GP.
5: **end for**

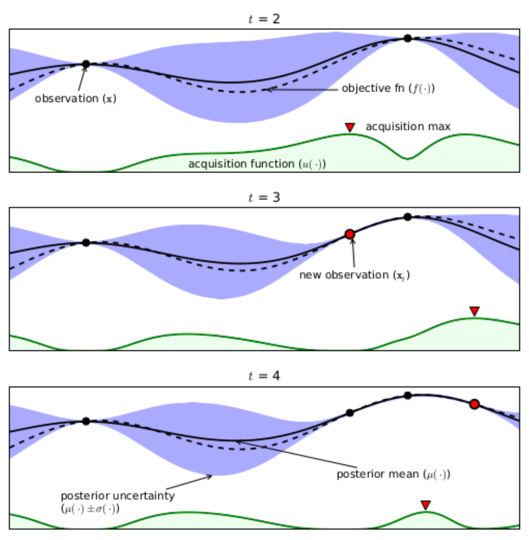
1. sample from function
2. choose next point based on acquisition function
3. repeat step 2 until converge

Using Uncertainty in Optimization

- target: find the maximum / minimum of f
- after performing some evaluations, the GP gives us means and variances
- next evaluation based on:
  exploration: points with high variance
  exploitation: points with high mean (max problem)
- acquisition function balances between exploration and exploitation.

BO in example

- dashed like: real objective function
- solid black line: posterior mean $\mu(x)$
- green line: acquisition function
- blue area: confidence area



Prior over Functions

Assume we have observations $(\boldsymbol{x}_{1:t}, \boldsymbol{f}_{1:t})$ and it follows the multivariate normal distribution $\mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$

Now, we have a new point $\boldsymbol{x}_{t+1}$, by properties of GP, $\boldsymbol{f}_{1:t}$ and f_{t+1} are jointly Gaussian:

$$\begin{pmatrix} \boldsymbol{f}_{1:t} \\ f_{t+1} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} \boldsymbol{K}, & \boldsymbol{k} \\ \boldsymbol{k}^\top, & k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \end{pmatrix}\right)$$

$\boldsymbol{k} = (k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t))$

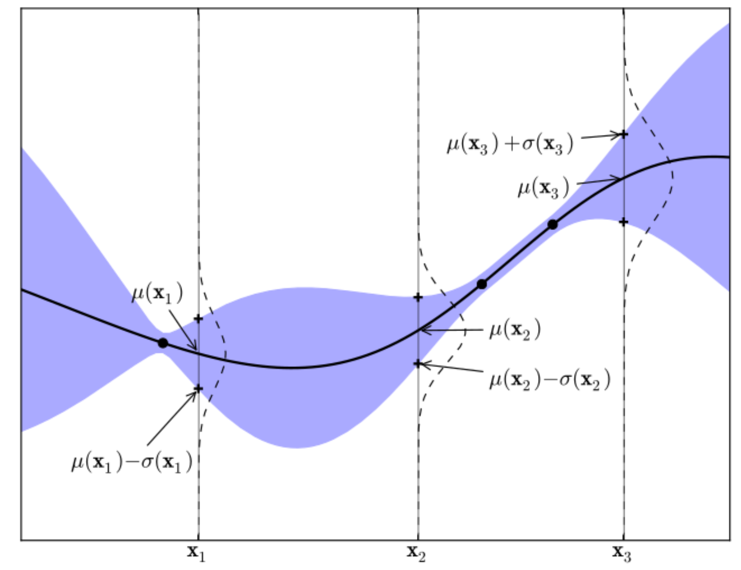
Then the predictive distribution:

$$p(f_{t+1}|\mathcal{D}_{1:t}, \boldsymbol{x}_{t+1}) = \mathcal{N}(\mu_t(\boldsymbol{x}_{t+1}), \sigma_t^2(\boldsymbol{x}_{t+1}))$$

$$\mu_t(\boldsymbol{x}_{t+1}) = \boldsymbol{k}^\top \boldsymbol{K}^{-1} \boldsymbol{f}_{1:t}$$

$$\sigma_t^2(\boldsymbol{x}_{t+1}) = k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) - \boldsymbol{k}^\top \boldsymbol{K}^{-1} \boldsymbol{k}$$

Prior over functions cont'd



Acquisition Functions

- Probability improvement

$$a_{PI}(\boldsymbol{x}) = \Phi\left(\frac{\mu(\boldsymbol{x}) - f_n^*}{\sigma(\boldsymbol{x})}\right)$$

- Expected improvement:

$$a_{EI}(\boldsymbol{x}) = \mathbb{E}\left[[f(\boldsymbol{x}) - f_n^*]^+\right]$$

- Upper confidence bound (UCB):

$$a_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x})$$

  where $\beta > 0$ is a tradeoff parameter.

- Entropy search / predicted entropy search:

$$a_{ES}(\boldsymbol{x}) = H(P(\boldsymbol{x}^*)) - \mathbb{E}_{f(\boldsymbol{x})}\left[H(P(\boldsymbol{x}^*|f(\boldsymbol{x})))\right]$$

$$a_{PES}(\boldsymbol{x}) = H(P(f(\boldsymbol{x}))) - \mathbb{E}_{\boldsymbol{x}^*}\left[H(P(f(\boldsymbol{x})|\boldsymbol{x}^*))\right]$$

Probability Improvement

$$
\begin{aligned}
a_{PI}(\boldsymbol{x}) =& p(f(\boldsymbol{x}) \geq f(\boldsymbol{x}^{+})) \\
=& \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^{+})}{\sigma(\boldsymbol{x})}\right)
\end{aligned}
$$

- still too greedy, pure exploitation

Sometimes modified as

$$
\begin{aligned}
a_{PI}(\boldsymbol{x}) =& p(f(\boldsymbol{x}) \geq f(\boldsymbol{x}^{+}) + \xi) \\
=& \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^{+}) - \xi}{\sigma(\boldsymbol{x})}\right)
\end{aligned}
$$

ξ is a tradeoff parameter.

Expected Improvement

$$a_{EI}(\boldsymbol{x}) = \mathbb{E}\left[[f(\boldsymbol{x}) - f_n^*]^+\right]$$

- $[\cdot]^+ \triangleq \max\{0, \cdot\}$ is a utility function
- $f_n^* \triangleq \min_{\boldsymbol{x}_{1:n}} f(\boldsymbol{x})$ is the current best

Integration by parts, we have the closed form:

$$a_{EI}(\boldsymbol{x}) = \begin{cases} (\mu(\boldsymbol{x}) - f_n^*)\Phi(Z) + \sigma(\boldsymbol{x})\phi(Z) & \sigma(\boldsymbol{x}) > 0 \\ 0 & \sigma(\boldsymbol{x}) = 0 \end{cases}$$

$$Z = \frac{\mu(\boldsymbol{x}) - f_n^*}{\sigma(\boldsymbol{x})}$$

$\Phi(Z)$ is the cumulative distribution and $\phi(Z)$ is the probability density function.

Upper Confidence Bound

$$a_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta\sigma(\boldsymbol{x})$$

Casting this as a multi-armed bandit, then the acquisition function is regret function:

$$r(\boldsymbol{x}) = f(\boldsymbol{x}^*) - f(\boldsymbol{x})$$

The goal is to find

$$\min \sum_{t}^{T} r(\boldsymbol{x}_t) = \max \sum_{t}^{T} f(\boldsymbol{x}_t)$$

Now select β as $\beta = \sqrt{\nu\tau_t}$, it can be shown with high probability this method is no regret

$$a_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sqrt{\nu\tau_t}\sigma(\boldsymbol{x})$$

[1] Srinivas et al, ICML 2010

Other Acquisition Functions

Entropy search / predicted entropy search:

$$a_{ES}(\boldsymbol{x}) = H(P(\boldsymbol{x}^*)) - \mathbb{E}_{f(\boldsymbol{x})}\left[H(P(\boldsymbol{x}^*|f(\boldsymbol{x})))\right]$$
$$a_{PES}(\boldsymbol{x}) = H(P(f(\boldsymbol{x}))) - \mathbb{E}_{\boldsymbol{x}^*}\left[H(P(f(\boldsymbol{x})|\boldsymbol{x}^*))\right]$$

Knowledge Gradient

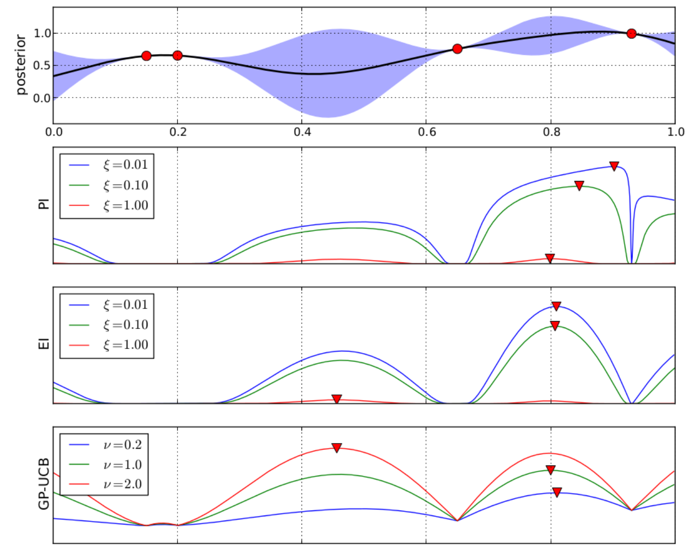
$$a_{KG}(\boldsymbol{x}) = \mathbb{E}\left[\mu^*_{n+1} - \mu^*_n | \boldsymbol{x}_{n+1} = \boldsymbol{x}\right]$$

Thompson Sampling

$$\int \mathbb{I}\left[\mathbb{E}(r|a^*, \boldsymbol{x}, \boldsymbol{\theta}) = \max \mathbb{E}(r|a^*, \boldsymbol{x}, \boldsymbol{\theta})\right] p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}$$

Taking the best a^* to maximize the reward.

Comparison of Acquisition Functions



Exotic Bayesian Optimization

What if the Bayesian optimization with following scenarios?

- Noisy evaluations
- Parallel evaluations
- Constraints
- Optimization of acquisition functions

Noisy Observations

If

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_{noise}^2)$$

Then the covariance becomes

$$cov(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma_{noise}^2 \delta_{ij}$$

where δ_{ij} is a Kronecker delta, then it yields the predictive distribution:

$$p(f_{t+1}|\mathcal{D}_{1:t}, \boldsymbol{x}_{t+1}) = \mathcal{N}(\mu_t(\boldsymbol{x}_{t+1}), \sigma_t^2(\boldsymbol{x}_{t+1}) + \sigma_{noise}^2)$$
$$\mu_t(\boldsymbol{x}_{t+1}) = \boldsymbol{k}^\top(\boldsymbol{K}^{-1} + \sigma_{noise}^2 \boldsymbol{I})\boldsymbol{f}_{1:t}$$
$$\sigma_t^2(\boldsymbol{x}_{t+1}) = k(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) - \boldsymbol{k}^\top(\boldsymbol{K}^{-1} + \sigma_{noise}^2 \boldsymbol{I})\boldsymbol{k}$$

Parallel

- Classical Bayesian optimization is sequential
  Do an experiment, wait until finishing, and repeat.
- Compute clusters let us do many things at once

$$a_{EI}(\boldsymbol{x}_{1:q}) = \mathbb{E}\left[\left[\max_{i=1..q} f(\boldsymbol{x}_i) - f^* n\right]^+\right]$$

- Fantasize outcomes from the GP
  GP gives coherent predictions and evaluate points that are good under the average
- Alternative: shrink the variance and integrate out the mean

[1] Desautels, et al, JMLR 2014

Constraints

What if the problem has complex constraints rather than a "bounded" simple domain?

$$\max f(\boldsymbol{x}) \quad \text{s.t.} c(\boldsymbol{x}) \leq 0$$

- evaluate the constraints separately from the objective
- evaluate the function only when the constraints are active
- a natural way following the "improvement" based acquisition functions

Optimization on Acquisition Functions

Ironic Problem:

Bayesian optimization has its own hyperparameters!

- covariance function has hyperparameters
- acquisition function has hyperparameters

How to attack them?

- Covariance hyperparameters are often optimized rather than marginalized, typically in the name of convenience and efficiency.
- Slice sampling of hyperparameters is fast and easy.
- Apply first- and second- order optimization methods.

Summary

Recent work

- Gaussian process v.s. RKHS in neural networks
  - *Deep Neural Networks as Gaussian Processes, ICLR 2018*
  - *Learning Transferable Features with Deep Adaptation Networks, JMLR 2015*
  - *Deep Kernel Learning, JMLR 2016*
- Gradient in Bayesian optimization
  - *Bayesian Optimization with Gradients, NIPS 2017*
  - *Do we need "harmless" bayesian optimization and first-order bayesian optimization, NIPS 2016*
- Optimization on acquisition function
  - *Maximizing acquisition functions for Bayesian optimization, arXiv 2018*
  - *The knowledge-gradient policy for correlated normal beliefs, J. of Comp. 2009*

Real Application

- Google's AutoML - use bayesian optimization
- AlphaGo Zero - self-play using Gaussian Process optimization (Bayesian optimization)
- Google Vizer - Internal use services to tune hyper-parameter
- Amazon SegaMaker

Applications including: robotics, automatic machine learning, hierarchical reinforcement learning.

Summary

- GPs define distributions on functions
- GPs are closely related to many other models:
  - Bayesian kernel machine
  - linear regression with basis functions
  - Deep neural networks
- GPs handle uncertainty in unknown function f by averaging
- BO performs the global optimizer over unknown functions
- BO helps the automatic machine learning
- BO has acquisition function to determine which point to pick in sequence.

Reference I

J. Bergstra and Y. Bengio.
Random search for hyper-parameter optimization.
Journal of Machine Learning Research, 13(Feb):281–305, 2012.

E. Brochu, V. M. Cora, and N. De Freitas.
A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.
arXiv preprint arXiv:1012.2599, 2010.

N. Fusi, R. Sheth, and H. M. Elibol.
Probabilistic matrix factorization for automated machine learning.
arXiv preprint arXiv:1705.05355, 2017.

Reference II

M. A. Gelbart, J. Snoek, and R. P. Adams.
Bayesian optimization with unknown constraints.
arXiv preprint arXiv:1403.5607, 2014.

P. Hennig and C. J. Schuler.
Entropy search for information-efficient global optimization.
Journal of Machine Learning Research, 13(Jun):1809–1837, 2012.

F. Hutter, H. H. Hoos, and K. Leyton-Brown.
Sequential model-based optimization for general algorithm configuration.
In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

Reference III

R. Lam and K. Willcox.
Lookahead bayesian optimization with inequality constraints.
In *Advances in Neural Information Processing Systems*, pages 1890–1900, 2017.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein.
Deep neural networks as gaussian processes.
arXiv preprint arXiv:1711.00165, 2017.

A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani.
Gaussian process behaviour in wide deep neural networks.
arXiv preprint arXiv:1804.11271, 2018.

Reference IV

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas.
Taking the human out of the loop: A review of bayesian optimization.
Proceedings of the IEEE, 104(1):148–175, 2016.

J. Snoek, H. Larochelle, and R. P. Adams.
Practical bayesian optimization of machine learning algorithms.
In *Advances in neural information processing systems*, pages 2951–2959, 2012.

J. Snoek, K. Swersky, R. Zemel, and R. Adams.
Input warping for bayesian optimization of non-stationary functions.
In *International Conference on Machine Learning*, pages 1674–1682, 2014.

Reference V

J. R. Snoek.
Bayesian optimization and semiparametric models with applications to assistive technology.
PhD thesis, University of Toronto, 2013.

K. Swersky, J. Snoek, and R. P. Adams.
Multi-task bayesian optimization.
In *Advances in neural information processing systems*, pages 2004–2012, 2013.

C. K. Williams and C. E. Rasmussen.
Gaussian processes for machine learning.
the MIT Press, 2006.

Reference VI

J. T. Wilson, F. Hutter, and M. P. Deisenroth.
Maximizing acquisition functions for bayesian optimization.
arXiv preprint arXiv:1805.10196, 2018.