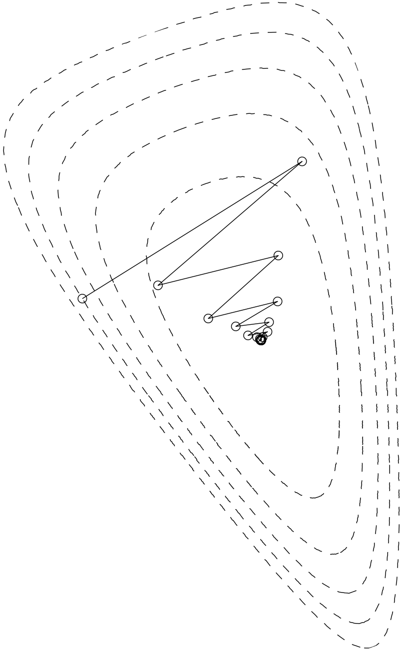


Optimization Algorithms

Bayesian Optimization

Marc Toussaint
Technical University of Berlin
Winter 2024/25

References

- *Information-theoretic regret bounds for gaussian process optimization in the bandit setting* Srinivas, Krause, Kakade & Seeger, Information Theory, 2012.
- *A taxonomy of global optimization methods based on response surfaces* Jones, Journal of Global Optimization, 2001.
- *Explicit local models: Towards optimal optimization algorithms*, Poland, Technical Report No. IDSIA-09-04, 2004.

Global Optimization

- Let $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x f(x)$$

- Blackbox optimization: find a global optimium by sampling values $y_t = f(x_t)$
  - No access to ∇f or $\nabla^2 f$
  - Observations may be noisy $y \sim \mathcal{N}(y \mid f(x_t), \sigma^2)$

Global Optimization

- Let $x \in \mathbb{R}^n$, $f : \ \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x \ f(x)$$

- Blackbox optimization: find a global optimium by sampling values $y_t = f(x_t)$
  - No access to ∇f or $\nabla^2 f$
  - Observations may be noisy $y \sim \mathcal{N}(y \mid f(x_t), \sigma^2)$

- Global Optimization = infinite Bandits, with infinite decision space, $x \in \mathbb{R}^n$
  - Bandit problems are archetype for sequential decision making under uncertainty
  - Upper Confidence Bound (UCB) decisions have provably bounded regret!
  - Resolves exploration/exploitation “dilemma”
  - Bayesian Optimization (GP-UCB) transfers bandits to continuous decisions $x \in \mathbb{R}^n$

Random Restarts (run downhill multiple times)

- first the most basic approach...

Random Restarts (run downhill multiple times)

- first the most basic approach...

- We assume to have a start distribution $q(x)$, and restart greedy search:

---

repeat
  Sample $x \sim q(x)$
  $x \leftarrow$ `GreedySearch`(x) or `StochasticSearch`(x)
  If $f(x) < f(x^*)$ **then** $x^* \leftarrow x$
until run out of budget

- When gradients are available, replace greedy search by BFGS or Newton

Random Restarts (run downhill multiple times)

- first the most basic approach...

- We assume to have a start distribution $q(x)$, and restart greedy search:

---

repeat
  Sample $x \sim q(x)$
  $x \leftarrow \texttt{GreedySearch}(x)$ or $\texttt{StochasticSearch}(x)$
  If $f(x) < f(x^*)$ **then** $x^* \leftarrow x$
until run out of budget

- When gradients are available, replace greedy search by BFGS or Newton

- Can we not *learn* more from all the evaluated points and found local optima?

Optimizing and Learning

- Blackbox optimization is often related to learning:
  - When we have local a gradient or Hessian, we can take that local information and run downhil – no need to keep track of the history or learn (exception: BFGS, momentum)
  - In the Blackbox case we have no local information directly accessible $\rightarrow$ one needs to account of the history in some way or another to have an idea where to continue search
- "Accounting for the history" often means learning or maintaining data:
  - Learning a local or global model of f itself, learning which steps have been successful recently (gradient estimation), or which step directions, or other heuristics
  - Maintaining data: populations, evolutionary algorithms, EDAs, etc.

- Where we left when discussing No Free Lunch:

  What are algorithms that literally start by making assumptions about $P(f)$ *and then derive an optimization algorithm for that* $P(f)$*?*

- In Bayesian Optimization we maintain a particular belief $b_t = P(f \mid D)$, namely a *Gaussian Process*, and choose the next query based on that.

Gaussian Processes

- In my ML lectures, I introduce Gaussian Processes as Bayesian Kernel Ridge Regression
  But here, the function space view of GPs relates more directly to NLF
  (see also Welling: "Kernel Ridge Regression" Lecture Notes; Rasmussen & Williams sections 2.1 & 6.2; Bishop sections 3.3.3 & 6)

Gaussian Process definition

- The function space view: We have a prior $P(f)$ and data, then

$$P(f|\text{Data}) = \frac{P(\text{Data}|f)\; P(f)}{P(\text{Data})}$$

- Gaussian Processes define a probability distribution over functions:
  - A function is an infinite dimensional thing – how could we define a Gaussian distribution over functions?
  - For every finite set $\{x_1, .., x_M\}$, the function values $f(x_1), .., f(x_M)$ are Gaussian distributed with mean and covariance

$$\mathbb{E}\{f(x_i)\} = \mu(x_i) \qquad \text{(often zero)}$$
$$\mathbb{E}\{[f(x_i) - \mu(x_i)][f(x_j) - \mu(x_j)]\} = k(x_i, x_j)$$

    where, $\mu(x)$ is called **mean function**, and $k(x, x')$ is called **covariance function**
  - μ and k generalize the notion of *mean vector* μ_x and *covariance matrix* $\Sigma_{xx'}$ from finite $x \in \{1, .., n\}$ to continuous $x \in \mathbb{R}^n$

- Second, Gaussian Processes define an observation probability

$$P(y|x, f) = \mathcal{N}(y|f(x), \sigma_0^2)$$

Gaussian Process posterior

- Given a Gaussian Process prior $GP(f|\mu, k)$ over f and data $D = \{(x_i, y_i)\}_{i=1}^n$, the posterior $P(f \mid D)$ has new posterior mean and variance:

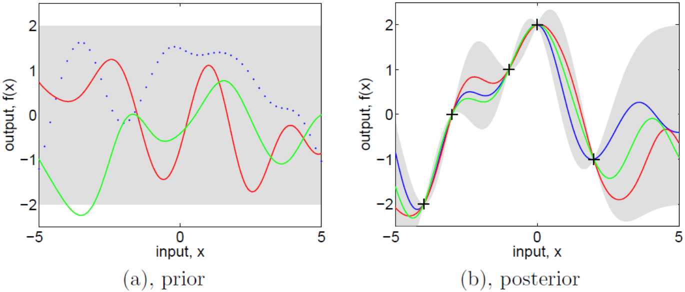
$$\mathbb{E}\{f(x) \mid D\} = \mu(x|D) = \kappa(x)^\top (K + \sigma_0^2 \mathbf{I})^{-1} y$$

$$\mathbb{E}\Big\{[f(x) - \hat f(x)]^2 \mid D\Big\} = \sigma^2(x|D) = k(x,x) - \kappa(x)^\top (K + \sigma_0^2 \mathbf{I}_n)^{-1} \kappa(x)$$

where $\kappa(x) = (k(x, x_1), \dots, k(x, x_n))^\top \in \mathbb{R}^n$ contains covariances of x to all data points; $K = (k(x_i, x_j))_{i,j=1}^{n,n}$ contains covariances between all data points; and $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ contains all data output values; the choice of kernel $k(\cdot, \cdot)$ and the observation sdv σ_0 are parameters

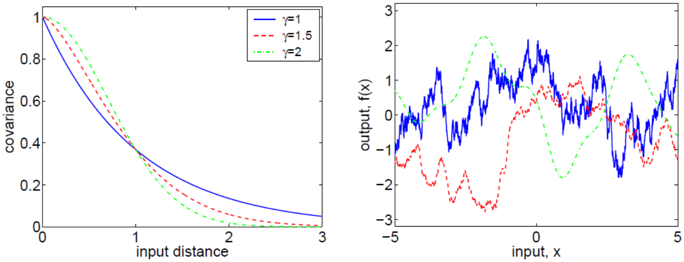
- Side notes:
  - Note: Don't forget that $\text{Var}(y^*|x^*, D) = \sigma_0^2 + \text{Var}(f(x^*)|D)$
  - Gaussian Processes = Bayesian Kernel Ridge Regression
  - GP classification = Bayesian Kernel Logistic Regression

GP examples



(from Rasmussen & Williams)

GP examples: different covariance functions



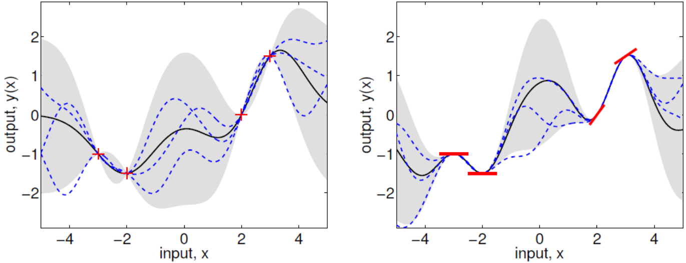
(from Rasmussen & Williams)

- These are examples from the γ-exponential covariance function

$$k(x, x') = a \ \exp\{-|(x - x')/l|^\gamma\}$$

with a the prior variance of function values

GP examples: derivative observations



(from Rasmussen & Williams)

Heuristics / Acquisition Functions

Bayesian Optimization Algorithm

Input: GP prior given as $\mu(x)$ and $k(x, x')$, black-box function $f(x)$
Output: x

1: initialize empty data $D = \{\}$
2: **repeat**
3: find optimal query $x \leftarrow \text{argmax}_x\, \alpha(x|D)$ (where α depends on $\mu(x|D), \sigma^2(x|D)$)
4: query $y \leftarrow f(x)$
5: add to data $D \leftarrow D \cup \{(x, y)\}$, update GP posterior $\mu(x|D), \sigma^2(x|D)$
6: **until** resources

- $\alpha(x; D)$ is called **acquisition function**
  - $\alpha(x; D)$ characterizes how "interesting" it is to query x next, given D
  - $\alpha(x; D)$ is a descriminative function for the next decision
  - $\alpha(x; D)$ analogous to a Q-function $Q(D, x)$ for the next decision x in state D

Acquisition Functions

- Maximize Probability of Improvement (MPI)

$$\alpha(x; D) = \int_{-\infty}^{y^*} \mathcal{N}(y|\mu_D(x), \sigma_D^2(x))$$

- Maximize Expected Improvement (EI)

$$\alpha(x; D) = \int_{-\infty}^{y^*} \mathcal{N}(y|\mu_D(x), \sigma_D^2(x))\ (y^* - y)$$

- Maximize UCB

$$\alpha(x; D) = \mu_D(x) + \beta_t \sigma_D^2(x)$$

(Often, $\beta_t = 1$ is chosen. UCB theory allows for better choices. See Srinivas et al. citation.)

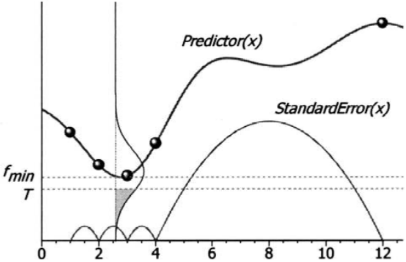


Figure 14. Using kriging, we can estimate the probability that sampling at a give
'improve' our solution, in the sense of yielding a value that is equal or better than
T.

(from Jones, 2001)

Each step requires solving an optimization problem

- Note: each $\text{argmax}_x\, \alpha(x)$ on the previous slide is an optimization problem!
- As $\mu(x|D), \sigma^2(x|D)$ are given analytically, we have gradients and Hessians. BUT: multi-modal problem!
- In practice:
  - Many restarts of gradient/2nd-order optimization runs
  - Restarts from a grid; from many random points

Each step requires solving an optimization problem

- Note: each $\mathrm{argmax}_x\, \alpha(x)$ on the previous slide is an optimization problem!
- As $\mu(x|D), \sigma^2(x|D)$ are given analytically, we have gradients and Hessians. BUT: multi-modal problem!
- In practice:
  - Many restarts of gradient/2nd-order optimization runs
  - Restarts from a grid; from many random points

- We traded a *blackbox* global optimization problem by solving an *analytical* global optimization problem in each iteration:
  - Assumes evaluating the real $f(x)$ is very expensive
  - The inner problem is analytical, can exploit gradients/Hessian, can run without real-world queries

GP-UCB

From: *Information-theoretic regret bounds for gaussian process optimization in the bandit setting* Srinivas, Krause, Kakade & Seeger, Information Theory, 2012.

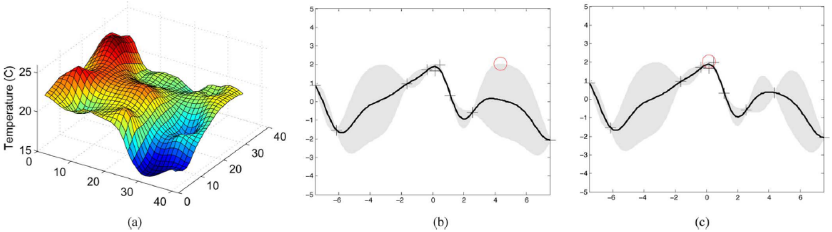


Fig. 2. (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b) and (c) Two iterations of the GP-UCB algorithm. The dark curve indicates the current posterior mean, while the gray bands represent the upper and lower confidence bounds which contain the function with high probability. The "+" mark indicates points that have been sampled before, while the "o" mark shows the point chosen by the GP-UCB algorithm to sample next. It samples points that are either (b) uncertain or have (c) high posterior mean.

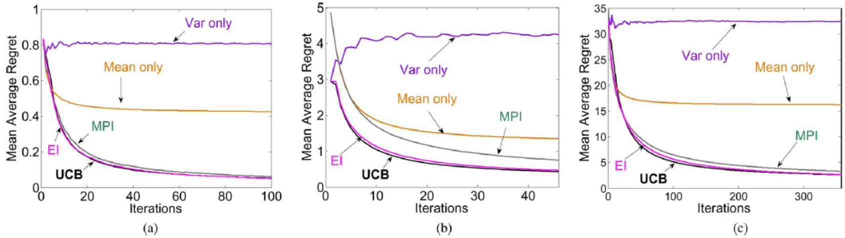


Fig. 6. Mean average regret: GP-UCB and various heuristics on (a) synthetic and (b, c) sensor network data.

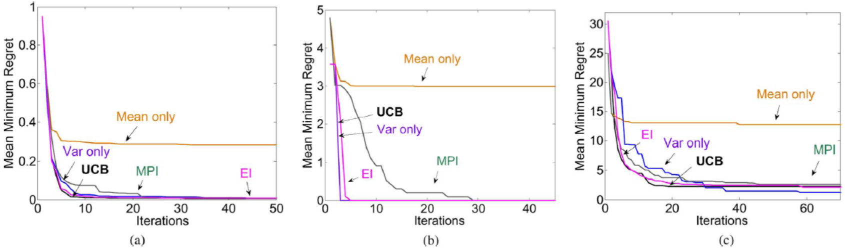


Fig. 7. Mean minimum regret: GP-UCB and various heuristics on (a) synthetic, and (b, c) sensor network data.

Pitfall of using GPs as belief

- A real issue, in my view, is the choice of kernel (i.e. prior $P(f)$)
  - 'small' kernel: almost exhaustive search
  - 'wide' kernel: miss local optima
  - adapting/choosing kernel online (with CV): might fail
  - real f might be non-stationary
  - non RBF kernels? Too strong prior, strange extrapolation

- Assuming that we have the right prior $P(f)$ is really a strong assumption

Further reading

- Classically, such methods are known as *Kriging*

- *Information-theoretic regret bounds for gaussian process optimization in the bandit setting* Srinivas, Krause, Kakade & Seeger, Information Theory, 2012.

- *Efficient global optimization of expensive black-box functions.* Jones, Schonlau, & Welch, Journal of Global Optimization, 1998.
- *A taxonomy of global optimization methods based on response surfaces* Jones, Journal of Global Optimization, 2001.
- *Explicit local models: Towards optimal optimization algorithms*, Poland, Technical Report No. IDSIA-09-04, 2004.

Further reading: Entropy Search

- P. Hennig & C. Schuler: *Entropy Search for Information-Efficient Global Optimization*, JMLR 13 (2012).

- **Predictive Entropy Search**
- Hernández-Lobato, Hoffman & Ghahraman: *Predictive Entropy Search for Efficient Global Optimization of Black-box Functions*, NIPS 2014.
- Also for constraints!
- Code: `https://github.com/HIPS/Spearmint/`

Note: beyond Gaussian Processes

- Use emsembles (e.g. bootstrap ensembles) of models and their discrepancy to decide on information gain, rather than variance!
  - Can be realized also with more complicated function models (NNs)
  - covariance function is implicit and more structured

Appendix

Other basic approaches...

Iterated Local Search

- Iterated Local Search (in discrete spaces) restarts in a **meta-neighborhood** $\mathcal{N}^*(x)$ of the last visited local minimum x
- Iterated Local Search (Variant 1):

Input: initial x, function $f(x)$

repeat

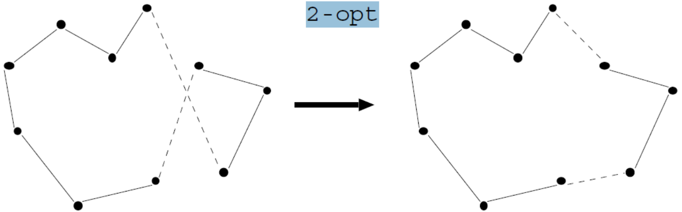
$\quad x \leftarrow \mathrm{argmin}_{y' \in \{\texttt{GreedySearch}(y)\,:\, y \in \mathcal{N}^*(x)\}} f(y')$

until x converges

  - This evalutes a GreedySearch for all meta-neighbors $y \in \mathcal{N}^*(x)$ of the last local optimum x
  - The inner GreedySearch uses another neighborhood function $\mathcal{N}(x)$
- Variant 2: $x \leftarrow$ the "first" $y \in \mathcal{N}^*(x)$ such that $f(\texttt{GS}(y)) < f(x)$
- In **continuous space**: $\mathcal{N}(x)$ and $\mathcal{N}^*(x)$ are replaced by transition proposals $q(y|x)$ and $q^*(y|x)$

Iterated Local Search

- Application to Travelling Salesman Problem:
  k-opt neighbourhood: solutions which differ by at most k edges



from Hoos & Stützle: *Tutorial: Stochastic Search Algorithms*

- GreedySearch uses 2-opt or 3-opt neighborhood
  Iterated Local Search uses 4-opt meta-neighborhood (double bridges)