Probabilistic Machine Learning
Lecture 25
Making Decisions

Philipp Hennig

20 July 2020

Eberhard Karls Universität Tübingen
Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

#	date	content	Ex	#	date	content	Ex
1	20.04.	Introduction	1	14	09.06.	Generalized Linear Models	
2	21.04.	Reasoning under Uncertainty		15	15.06.	Exponential Families	8
3	27.04.	Continuous Variables	2	16	16.06.	Graphical Models	
4	28.04.	Monte Carlo		17	22.06.	Factor Graphs	9
5	04.05.	Markov Chain Monte Carlo	3	18	23.06.	The Sum-Product Algorithm	
6	05.05.	Gaussian Distributions		19	29.06.	Example: Modelling Topics	10
7	11.05.	Parametric Regression	4	20	30.06.	Mixture Models	
8	12.05.	Learning Representations		21	06.07.	EM	11
9	18.05.	Gaussian Processes	5	22	07.07.	Variational Inference	
10	19.05.	Understanding Kernels		23	13.07.	Tuning Inference Algorithms	12
11	26.05.	Gauss-Markov Models		24	14.07.	Kernel Topic Models	
12	25.05.	An Example for GP Regression	6	25	20.07.	Outlook	
13	08.06.	GP Classification	7	26	21.07.	Revision	

The Toolbox

Framework:

$$\int p(x_1, x_2)\, dx_2 = p(x_1) \qquad p(x_1, x_2) = p(x_1 \mid x_2) p(x_2) \qquad p(x \mid y) = \frac{p(y \mid x) p(x)}{p(y)}$$

Modelling:

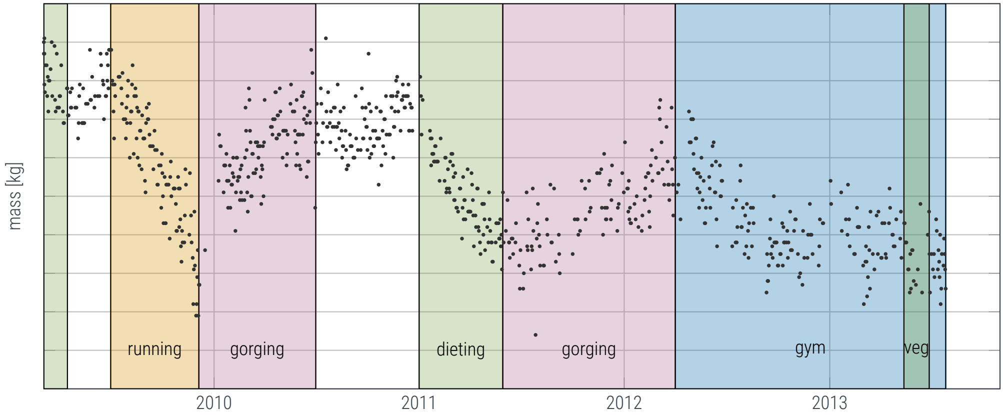
- graphical models
- Gaussian distributions
- (deep) learnt representations
- Kernels
- Markov Chains
- Exponential Families / Conjugate Priors
- Factor Graphs & Message Passing

Computation:

- Monte Carlo
- Linear algebra / Gaussian inference
- maximum likelihood / MAP
- Laplace approximations
- EM / variational approximations

So you've got yourself a posterior …now what?

Taking a decision means *conditioning* on a variable you control



$$p(w' \mid \text{run}) \qquad p(w' \mid \text{diet})$$

Decision Theory

The limit of probabilistic reasoning?

- probabilistic models can provide predictions $p(x \mid a)$ for a variable x *conditional* on an action a
- given the choice, which value of a do you prefer?

Decision Theory

The limit of probabilistic reasoning?

- probabilistic models can provide predictions $p(x \mid a)$ for a variable x *conditional* on an action a
- given the choice, which value of a do you prefer?

- assign a *loss* or *utility* $\ell(x)$
- choose a such that it minimizes expected loss

$$a_* = \arg\min_a \int \ell(x) p(x \mid a)\, dx$$

Expected Regret/utility

if you keep having to take the same decision, optimise the sum of its return

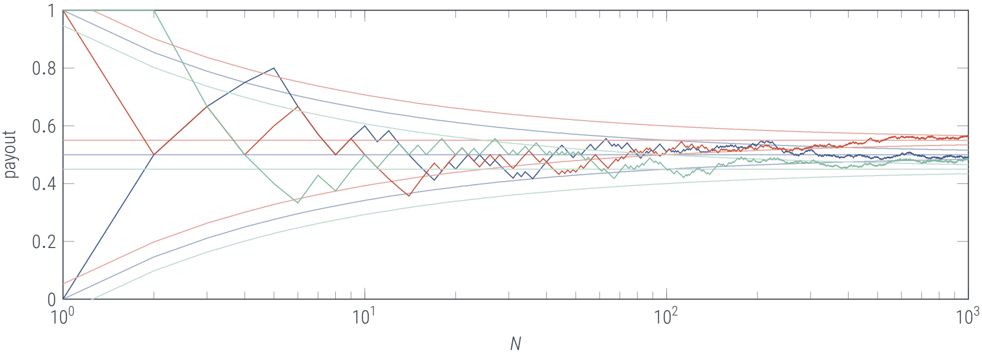
- consider *independent* draws x_i with $x_i \sim p(x \mid a_i)$
- choose all $a_i = a_*$ to minimize the accumulated loss

$$L(n) = \mathbb{E}_p \left[\sum_i x_i \right]$$

- but what if you *don't know* p?

Motivating (Historical) Example

Experimental Design



Perhaps we shouldn't rule out an option yet if the posteriors over their expected return overlaps with that of our current guess for the best option?

- Assume K choices.
- Taking choice $k \in [1, \dots, K]$ at time i yields binary (Bernoulli) reward/loss x_i with probability $\pi_k \in [0, 1]$, iid.
- conjugate priors $p(\pi_k) = \mathcal{B}(\pi, a, b) = B(a, b)^{-1}\pi^{a-1}(1-\pi)^{b-1}$
- posteriors from n_k trys of choice k with m_k successes: $p(\pi_k \mid n_k, m_k) = \mathcal{B}(\pi_k; a + m_k, b + (n_k - m_k))$
- for $a, b \rightarrow 0$, posterior has mean and variance

$$\bar{\pi}_k := \mathbb{E}_{p(\pi_k \mid n_k, m_k)}[\pi] = \frac{m_k}{n_k} \qquad \sigma_k^2 := \operatorname{var}_{p(\pi_k \mid n_k, m_k)}[\pi] = \frac{m_k(n_k - m_k)}{n_k^2(n_k + 1)} = \mathcal{O}(n_k^{-1})$$

Choose option k that maximizes $\bar{\pi}_k + c\sqrt{\sigma_k^2}$ for some c. Which c?

Learning by Doing

Estimating return while taking actions

Perhaps we shouldn't rule out an option yet if the posteriors over their expected return overlaps with that of our current guess for the best option?

Choose option k that maximizes $\bar{\pi}_k + c\sqrt{\sigma_k^2}$ for some c. Which c?

- A large c ensures uncertain options are preferred. If we make it too large, we will only *explore*.
- A small c largely ignores uncertainty. We will only *exploit*.
- Idea: Let c grow slowly over time, at rate less than $\mathcal{O}(n_k^{1/2})$. Then variance of chosen options will drop faster than c grows, so their exploration will stop, unless their mean is good. But unexplored choices will eventually become dominant, thus always explored eventually.

Not just for Bernoulli variables!

posterior contraction rates are universal

Theorem (Chernoff-Hoeffding)

Let $X_1, \dots, X_n$ be random variables with common range $[0, 1]$ and such that $\mathbb{E}[X_t \mid X_1, \dots, X_{t-1}] = \mu$. Let $S_n = X_1 + \dots + X_n$. Then for all $a \geq 0$,

$$p(S_n - n\mu \leq -a) \leq e^{-2a^2/n} \qquad \textit{and} \qquad p(S_n - n\mu \geq a) \leq e^{-2a^2/n}$$

The Multi-Armed Bandit Setting

Discrete-Choice Experimental Design

[Auer, Cesa-Bianchi, Fischer, Machine Learning **47**(2002), 235–256]

Definitions:

- A K-**armed bandit** is a collection X_{kn} of random variables, $1 \leq k \leq K$, $n \geq 1$ where k is the arm of the bandit. Successive plays of k yield rewards $X_{k1}, X_{k2}, \ldots$ which are **independent and identically distributed** according to an unknown p with $\mathbb{E}_p(X_{ki}) = \mu_i$.
- A **policy** A chooses the next machine to play at time n, based on past plays and rewards.
- Let $T_k(n)$ be number of times machine k was played by A during the first n plays. The **regret** of A is

$$R_A(n) = \mu^* \cdot n - \sum_j \mu_j \cdot \mathbb{E}_p[T_j(n)] \qquad \text{with } \mu^* := \max_{1 \leq k \leq K} \mu_k$$

The Multi-Armed Bandit Setting

Discrete-Choice Experimental Design

[Auer, Cesa-Bianchi, Fischer, Machine Learning **47**(2002), 235–256]

Algorithm: Let $\bar{x}_j$: empirical average of rewards from j, n_j: number of plays at j in n plays

procedure UCB(K) // **U**pper **C**onfidence **B**ound

  play each machine once

  while true **do**

    play $j = \arg\max \left(\bar{x}_j + \sqrt{\frac{2 \log n}{n_j}} \right)$

  end while

end procedure

The Multi-Armed Bandit Setting

Discrete-Choice Experimental Design

[Auer, Cesa-Bianchi, Fischer, Machine Learning **47**(2002), 235–256]

Theorem (Auer, Cesa-Bianchi, Fischer)

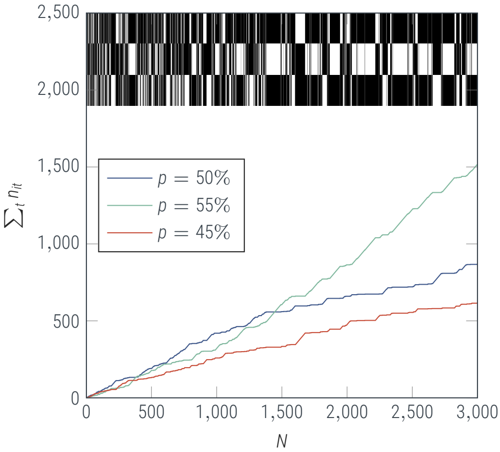
Consider K machines ($K > 1$) having ***arbitrary*** *reward distributions $P_1, \dots, P_K$ with support in $[0, 1]$ and expected values $\mu_i = \mathbb{E}_P(X_i)$. Let $\Delta_i := \mu_* - \mu_i$. Then, the expected regret of UCB after any number n of plays is at most*

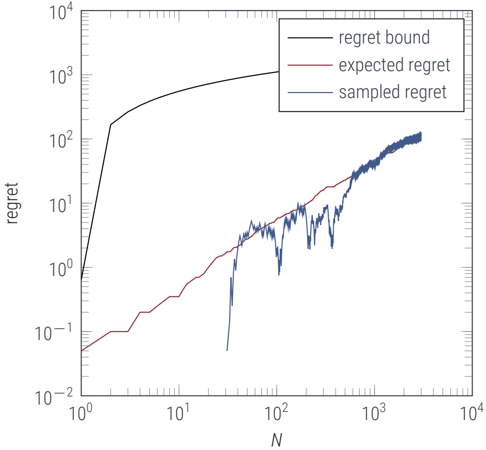
$$\mathbb{E}_P[R_A(n)] \leq \left[8 \sum_{i:\mu_i \leq \mu^*} \left(\frac{\log n}{\Delta_i} \right) \right] + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_j \Delta_j \right)$$

Nb: The sums are over K, not n. So the regret is $\mathcal{O}(K \log n)$. UCB plays a sub-optimal arm at most logarithmically often.

Visualization

$K = 3$, binary rewards





Multi-Armed Bandit Algorithms

- apply to *independent, discrete* choice problems with stochastic pay-off
- algorithms based on upper confidence bounds incur regret bounded by $\mathcal{O}(\log n)$
- this even applies for the *adversarial* setting (Auer, Cesa-Bianchi, Freund, Schapire, 1995)

Multi-Armed Bandit Algorithms

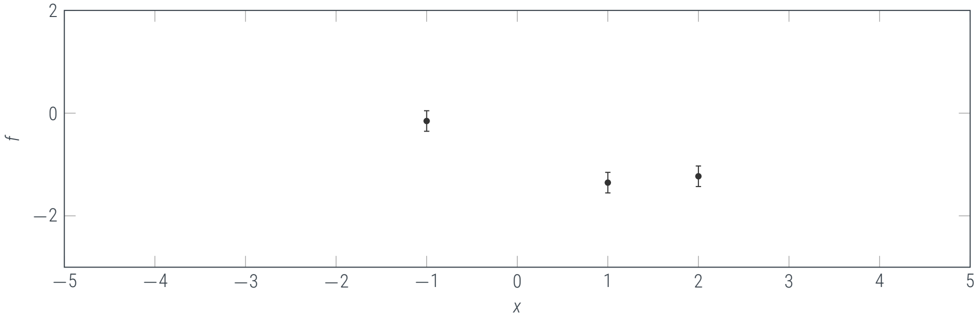
- apply to *independent, discrete* choice problems with stochastic pay-off
- algorithms based on upper confidence bounds incur regret bounded by $\mathcal{O}(\log n)$
- this even applies for the *adversarial* setting (Auer, Cesa-Bianchi, Freund, Schapire, 1995)

Unfortunately...

- No problem is ever discrete, finite and independent
- in a continuous problem, no “arm” can and should ever be played twice
- in many prototyping settings, early exploration is free

Continuous-Armed Bandits

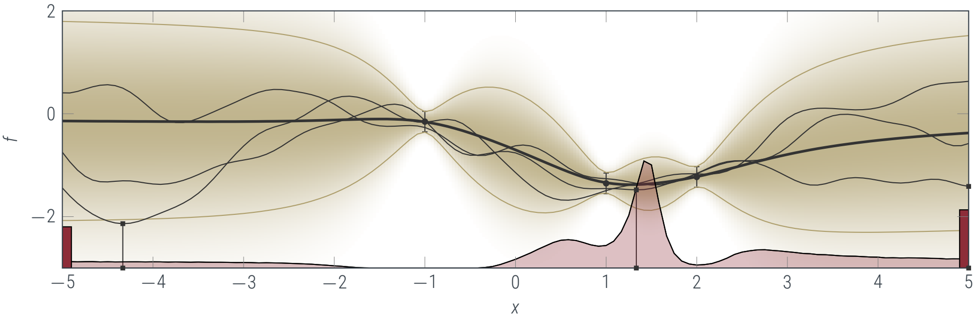
example application: parameter optimization



$$p(y \mid x) = \mathcal{N}(y; f_x, \sigma^2) \qquad x_* = \underset{x \in \mathbb{D}}{\arg\min}\, f(x) = ? \qquad R(T) := \sum_{t=1}^{T} f(x_t) - f(x_*)$$

Continuous-Armed Bandits

example application: parameter optimization

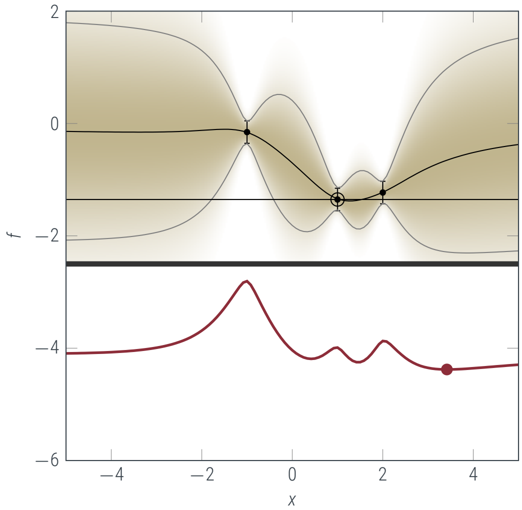


$$p(y \mid x) = \mathcal{N}(y; f_x, \sigma^2) \qquad p(f) = \mathcal{GP}(f; \mu, k) \quad \Rightarrow \quad p_{\min}(x_* = x) = \int_{\mathbb{R}} \int_{\mathbb{D}} \mathbb{I}(f(x) < f(\tilde{x}))\, d\tilde{x}\, dp(f \mid y)$$

GP Upper Confidence Bound

Evaluate *optimistally*, where the function *may* be low

Srinivas, Krause, Kakade, Seeger, ICML 2009



- **utility** under $p(f \mid y) = \mathcal{GP}(f; \mu_{t-1}, \sigma^2_{t-1})$

$$u_i(x) = \mu_{i-1}(x) - \sqrt{\beta_t}\sigma_{t-1}(x)$$

- choose x_t as $x_t = \arg\min_{x \in \mathbb{D}} u(x)$

Theorem (Srinivas et al., 2009)

*Let $\delta \in (0,1)$ and $\beta_t = 2\log(|\mathbb{D}|t^2\pi^2/6\delta)$. Running GP-UCB with β_t for **a sample** $f \sim GP(\mu, k)$,*

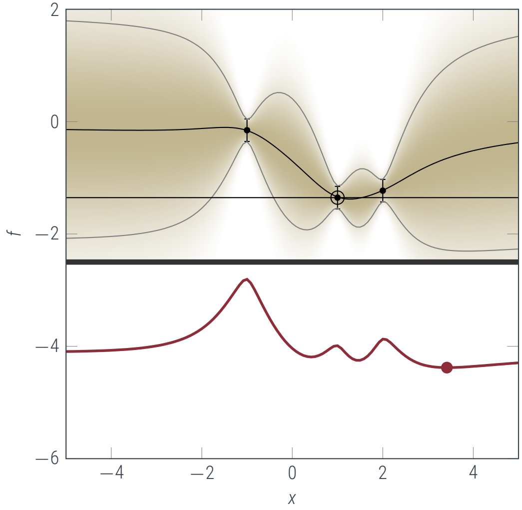
$$p\left(R_T \leq \sqrt{8T\beta_T\gamma_T/\log(1+\sigma^2)}\ \forall T \geq 1\right) \geq 1 - \delta$$

thus $\lim_{T\to\infty} R_T/T = 0$ ("no regret").

GP Upper Confidence Bound

Evaluate *optimistically*, where the function *may* be low

Srinivas, Krause, Kakade, Seeger, ICML 2009



- **utility** under $p(f \mid y) = \mathcal{GP}(f; \mu_{t-1}, \sigma^2_{t-1})$

$$u_i(x) = \mu_{i-1}(x) - \sqrt{\beta_t}\sigma_{t-1}(x)$$

- choose x_t as $x_t = \arg\min_{x \in \mathbb{D}} u(x)$

Theorem (Srinivas et al., 2009)

Assume that $f \in \mathcal{H}_k$ with $\|f\|_k^2 \leq B$, and the noise is zero-mean and σ-bounded almost surely. Let $\delta \in (0,1)$ and $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$. Running GP-UCB with β_t and $p(f) = \mathcal{GP}(f; 0, k)$,

$$p\left(R_T \leq \sqrt{8T\beta_T\gamma_T / \log(1+\sigma^2)} \; \forall T \geq 1\right) \geq 1-\delta$$

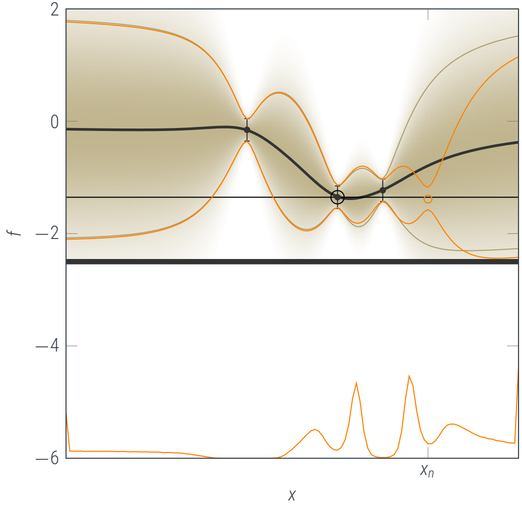
thus $\lim_{T\to\infty} R_T/T = 0$ ("no regret").

What if you have budget for several experiments?

Entropy Search

evaluate where you *expect to learn most about the minimum*

[Villemonteix et al., 2009; Hennig & Schuler, 2012]



- $p(f) = \mathcal{GP}(f; m, k)$ and
  $p(y \mid f) = \mathcal{N}(y; f_x, \sigma^2)$ gives
  $p(f \mid y) = \mathcal{N}(f; \mu, k)$, and

$$\bar{\mu}_a = \mu_a + \kappa_{a*}\kappa_{**}^{-1}(y_* - \mu_*)$$
$$= \mu_a + \underbrace{\kappa_{a*}\kappa_{**}^{-1/2}}_{=:L_{a*}} \cdot \underbrace{\kappa_{**}^{-1/2}(y_* - \mu_*)}_{u \sim \mathcal{N}(0,I)}$$

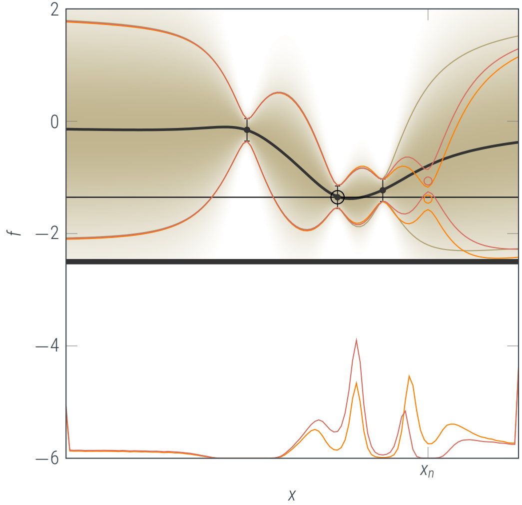
$$\bar{\kappa}_{ab} = \kappa_{ab} - \kappa_{a*}\kappa_{**}^{-1}\kappa_{*b}$$
$$= \kappa_{ab} - L_{a*}L_{*b}$$

- use this to predict $\hat{p}_{\min}(x)$ under $p(f \mid y, y_{t+1})$ (requires nontrivial numerics)

Entropy Search

evaluate where you *expect to learn most about the minimum*

[Villemonteix et al., 2009; Hennig & Schuler, 2012]



- $p(f) = \mathcal{GP}(f; m, k)$ and
  $p(y \mid f) = \mathcal{N}(y; f_x, \sigma^2)$ gives
  $p(f \mid y) = \mathcal{N}(f; \mu, k)$, and

$$\begin{aligned}\bar{\mu}_a &= \mu_a + \kappa_{a*}\kappa_{**}^{-1}(y_* - \mu_*) \\ &= \mu_a + \underbrace{\kappa_{a*}\kappa_{**}^{-1/2}}_{=:L_{a*}} \cdot \underbrace{\kappa_{**}^{-1/2}(y_* - \mu_*)}_{u \sim \mathcal{N}(0,I)}\end{aligned}$$

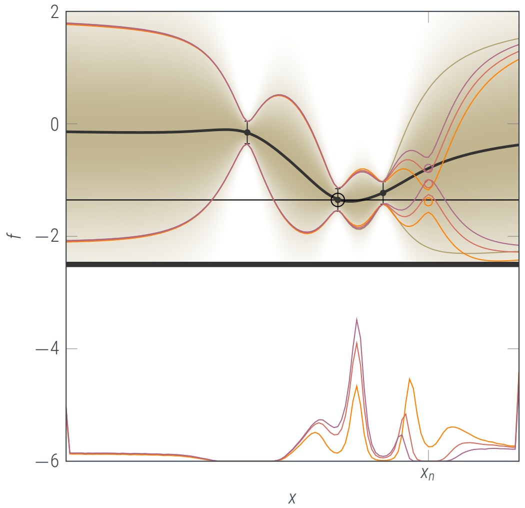
$$\begin{aligned}\bar{\kappa}_{ab} &= \kappa_{ab} - \kappa_{a*}\kappa_{**}^{-1}\kappa_{*b} \\ &= \kappa_{ab} - L_{a*}L_{*b}\end{aligned}$$

- use this to predict $\hat{p}_{\min}(x)$ under $p(f \mid y, y_{t+1})$ (requires nontrivial numerics)

Entropy Search

evaluate where you *expect to learn most about the minimum*

[Villemonteix et al., 2009; Hennig & Schuler, 2012]



- $p(f) = \mathcal{GP}(f; m, k)$ and
  $p(y \mid f) = \mathcal{N}(y; f_x, \sigma^2)$ gives
  $p(f \mid y) = \mathcal{N}(f; \mu, k)$, and

$$\bar{\mu}_a = \mu_a + \kappa_{a*}\kappa_{**}^{-1}(y_* - \mu_*)$$
$$= \mu_a + \underbrace{\kappa_{a*}\kappa_{**}^{-1/2}}_{=:L_{a*}} \cdot \underbrace{\kappa_{**}^{-1/2}(y_* - \mu_*)}_{u\sim\mathcal{N}(0,I)}$$

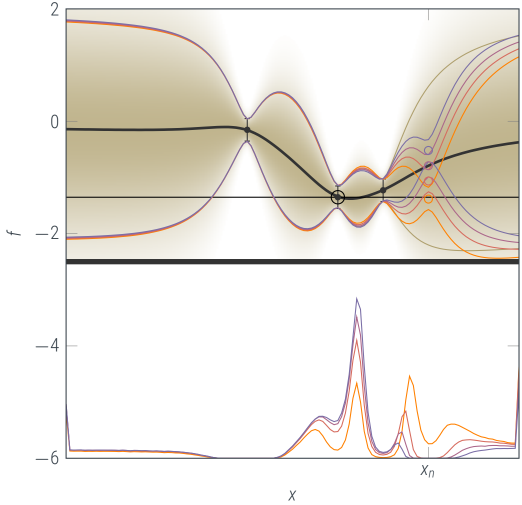
$$\bar{\kappa}_{ab} = \kappa_{ab} - \kappa_{a*}\kappa_{**}^{-1}\kappa_{*b}$$
$$= \kappa_{ab} - L_{a*}L_{*b}$$

- use this to predict $\hat{p}_{\min}(x)$ under $p(f \mid y, y_{t+1})$ (requires nontrivial numerics)

Entropy Search

evaluate where you *expect to learn most about the minimum*

[Villemonteix et al., 2009; Hennig & Schuler, 2012]

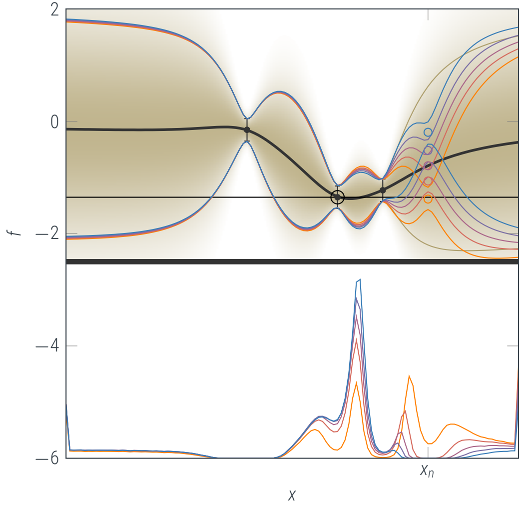


- $p(f) = \mathcal{GP}(f; m, k)$ and
  $p(y \mid f) = \mathcal{N}(y; f_x, \sigma^2)$ gives
  $p(f \mid y) = \mathcal{N}(f; \mu, k)$, and

$$\bar{\mu}_a = \mu_a + \kappa_{a*}\kappa_{**}^{-1}(y_* - \mu_*)$$
$$= \mu_a + \underbrace{\kappa_{a*}\kappa_{**}^{-1/2}}_{=:L_{a*}} \cdot \underbrace{\kappa_{**}^{-1/2}(y_* - \mu_*)}_{u\sim\mathcal{N}(0,I)}$$

$$\bar{\kappa}_{ab} = \kappa_{ab} - \kappa_{a*}\kappa_{**}^{-1}\kappa_{*b}$$
$$= \kappa_{ab} - L_{a*}L_{*b}$$

- use this to predict $\hat{p}_{\min}(x)$ under $p(f \mid y, y_{t+1})$ (requires nontrivial numerics)



- $p(f) = \mathcal{GP}(f; m, k)$ and
  $p(y \mid f) = \mathcal{N}(y; f_x, \sigma^2)$ gives
  $p(f \mid y) = \mathcal{N}(f; \mu, k)$, and

$$\begin{aligned}\bar{\mu}_a &= \mu_a + \kappa_{a*}\kappa_{**}^{-1}(y_* - \mu_*)\\ &= \mu_a + \underbrace{\kappa_{a*}\kappa_{**}^{-1/2}}_{=:L_{a*}} \cdot \underbrace{\kappa_{**}^{-1/2}(y_* - \mu_*)}_{u \sim \mathcal{N}(0,I)}\end{aligned}$$

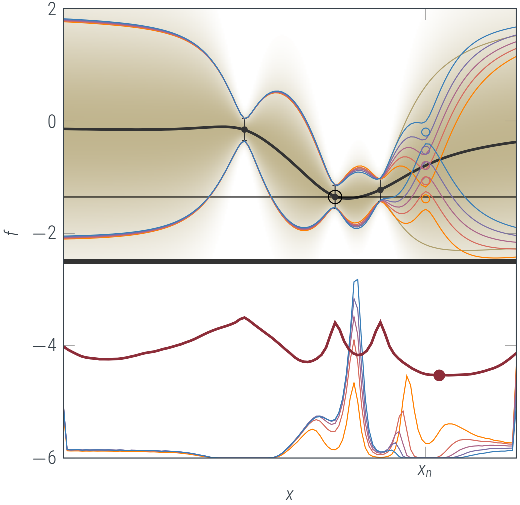
$$\begin{aligned}\bar{\kappa}_{ab} &= \kappa_{ab} - \kappa_{a*}\kappa_{**}^{-1}\kappa_{*b}\\ &= \kappa_{ab} - L_{a*}L_{*b}\end{aligned}$$

- use this to predict $\hat{p}_{\min}(x)$ under $p(f \mid y, y_{t+1})$ (requires nontrivial numerics)

Entropy Search

evaluate where you *expect to learn most about the minimum*

[Villemonteix et al., 2009; Hennig & Schuler, 2012]



- don't evaluate where you think the minium lies!
- instead, evaluate where you **expect to learn most about the minimum!**

$$\mathbb{H}(p) := -\int p(x) \log \frac{p(x)}{b(x)}\, dx$$

with *base measure b*. Use utility

$$u(x) = \mathbb{H}_t(p_{\min}) - \mathbb{E}_{y_{t+1}}[\mathbb{H}_{t+1}(p_{\min})]$$

Information vs. Regret

Entropy Search is qualitatively different from regret-based formulations

Settings in which information-based search is preferrable

- “prototyping-phase” followed by “product release”
- structured uncertainty with variable signal-to-noise ratio
- “multi-fidelity”: Several experimental channels of different cost and quality, e.g.
  - simulations vs. physical experiments
  - training a learning model for a variable time
  - using variable-size datasets

Regret-based optimization is easy to implement and works well on standard problems. But it is a strong simplification of reality, in which many pratical complications can not be phrased.

Bayesian Optimization in Practice

recent (and not so recent) libraries

- `https://amzn.github.io/emukit/`
- `https://github.com/HIPS/Spearmint`
- `https://github.com/hyperopt`
- `https://hpolib.readthedocs.io/en/development/`
- `https://github.com/automl`
- `https://sigopt.com/product/`

Summary — Experimental Design

- the **bandit setting** formalizes iid. sequential decision making under uncertainty
- bandit algorithms can achieve “no regret” performance, even without explicit probabilistic priors
- **Bayesian optimization** extends to continuous domain
- it lies right at the intersection of computational and physical learning
- requires significant computational resources to run *a numerical optimizer inside the loop*
- allows rich formulation of global, stochastic, continuous, structured, multi-channel design problems
- is currently the state of the art in the solution of challenging optimization problems