STAT G8325
Gaussian Processes and Kernel Methods
Lecture Notes §07:
Bayesian Optimization and Active Learning

John P. Cunningham

Department of Statistics
Columbia University

Outline

Outline

Administrative interlude

Bayesian optimization

Bayesian active learning

References

Progress...

Week	Lectures	Content
X	Oct 26	Special guest lecture by Andrew Gelman
X	Oct 28	No lecture (Cunningham unavailable)
X	Nov 2	No lecture (University holiday)
7	Nov 4,9	Bayesian optimization and active learning • [SLA12]; [GSW$^+$15]; [HHGL11]
8	Nov 9, 11	Kernel theory: existence, reproducing kernel Hilbert spaces, etc. • [Wah90, ch. 1] (intentionally light reading; work on projects)

- HW3 due end of this week.
- Lighter reading going forward.
- Transitioning into kernel methods (non gp).

Outline

Expensive evaluations

- Many core methods have (a few) tunable parameters.

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell, \gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i = j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\begin{aligned} &\text{maximize}_\alpha \qquad y_\alpha(\ell,\gamma) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j w_i w_j \left(k_\ell(z_i,z_j) + \gamma\mathbb{I}(i=j)\right) \\ &\text{subject to} \qquad \sum_{i=1}^n \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0. \end{aligned}$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\begin{aligned} \text{maximize}_\alpha \qquad & y_\alpha(\ell,\gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i=j)\right) \\ \text{subject to} \qquad & \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0. \end{aligned}$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\begin{aligned} &\text{maximize}_\alpha && y_\alpha(\ell,\gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i=j)\right) \\ &\text{subject to} && \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0. \end{aligned}$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_{\alpha} \qquad y_\alpha(\ell, \gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i = j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^*

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell, \gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i = j) \right)$$

$$\text{subject to} \qquad \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^* (recall this is the dual).

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell,\gamma) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j w_i w_j \left(k_\ell(z_i,z_j) + \gamma\mathbb{I}(i=j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^n \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell,\gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell,\gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^* (recall this is the dual).
- y_i expensive to evaluate

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell,\gamma) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j w_i w_j \left(k_\ell(z_i,z_j) + \gamma\mathbb{I}(i=j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^n \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^* (recall this is the dual).
- y_i expensive to evaluate $\rightarrow$ brute-force search is badly inefficient.

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell, \gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i = j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^{n} \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^* (recall this is the dual).
- y_i expensive to evaluate $\rightarrow$ brute-force search is badly inefficient.
- Alternatives to brute-force are the 'art' of applied stats

Expensive evaluations

- Many core methods have (a few) tunable parameters.
- Example: kernel bandwidth ℓ and slack γ in a soft-margin kernel SVM:

$$\text{maximize}_\alpha \qquad y_\alpha(\ell,\gamma) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j w_i w_j \left(k_\ell(z_i, z_j) + \gamma \mathbb{I}(i=j)\right)$$

$$\text{subject to} \qquad \sum_{i=1}^n \alpha_i w_i = 0 \quad , \quad \alpha_i \geq 0.$$

- Cross-validation to find optimal $x = [\ell, \gamma]$:
  - Grid the space $\mathcal{X}$ of reasonable parameter values $x = [\ell, \gamma]$.
  - Train and validate the SVM on each candidate x_i, producing y_i.
  - Choose the point x^* with largest y^* (recall this is the dual).
- y_i expensive to evaluate $\rightarrow$ brute-force search is badly inefficient.
- Alternatives to brute-force are the 'art' of applied stats $\rightarrow$ sloppy science.

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...
- Then assume $f \sim \mathcal{GP}(0, k)$...

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...
- Then assume $f \sim \mathcal{GP}(0, k)$... *a surrogate model*.

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...
- Then assume $f \sim \mathcal{GP}(0, k)$... *a surrogate model*.
- *Bayesian optimization (BO): Exploit a comparatively cheap gp surrogate model f to make smarter decisions about where to evaluate the true (very expensive to evaluate) function of interest y.*

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...
- Then assume $f \sim \mathcal{GP}(0, k)$... *a surrogate model*.
- *Bayesian optimization (BO): Exploit a comparatively cheap gp surrogate model f to make smarter decisions about where to evaluate the true (very expensive to evaluate) function of interest y.*
- Intuitively, the gp (via the smoothness of its kernel) gives insights as to global properties of the function, notably its extrema.

Assumption and idea

- Points y_i are noisy observations of an unknown function f to be optimized.
- If f is reasonably behaved (approximately smooth, bounded, etc)...
- Then assume $f \sim \mathcal{GP}(0, k)$... *a surrogate model*.
- *Bayesian optimization (BO): Exploit a comparatively cheap gp surrogate model f to make smarter decisions about where to evaluate the true (very expensive to evaluate) function of interest y.*
- Intuitively, the gp (via the smoothness of its kernel) gives insights as to global properties of the function, notably its extrema.
- Fairly old idea [MTZ78, Jon01]; more recently [SKKS09, SLA12].

The model

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.
- Key concept: cheaply (grid) search over $p(f|D)$ to find the next x_{i+1}.

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.
- Key concept: cheaply (grid) search over $p(f|D)$ to find the next x_{i+1}.
- But how to pick a good candidate from a posterior distribution?

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.
- Key concept: cheaply (grid) search over $p(f|D)$ to find the next x_{i+1}.
- But how to pick a good candidate from a posterior distribution?
- This search is judged by the *acquisition function* $a : \mathcal{X} \to \mathbb{R}$:

$$x_{i+1} \quad = \quad \arg\max_x \quad a\left(x|D, \theta\right).$$

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.
- Key concept: cheaply (grid) search over $p(f|D)$ to find the next x_{i+1}.
- But how to pick a good candidate from a posterior distribution?
- This search is judged by the *acquisition function* $a : \mathcal{X} \to \mathbb{R}$:

$$x_{i+1} \quad = \quad \arg\max_x \quad a\left(x|D, \theta\right).$$

- Critical: result depends substantially on $a(\cdot)$

The model

- Loss function: $f \sim \mathcal{GP}(0, k_\theta)$.
- Observations: $y_i | x_i \sim \mathcal{N}(x_i, \sigma_\epsilon^2)$.
- Goal: $\min_x \; y(x)$.
- Key concept: cheaply (grid) search over $p(f|D)$ to find the next x_{i+1}.
- But how to pick a good candidate from a posterior distribution?
- This search is judged by the *acquisition function* $a : \mathcal{X} \rightarrow \mathbb{R}$:

$$x_{i+1} = \arg\max_x \; a\left(x|D, \theta\right).$$

- Critical: result depends substantially on $a(\cdot)$ ...(and the kernel hypers θ).

Acquisition function: a feature of a distribution

Acquisition function: a feature of a distribution

- **Minimum mean**:

$$a(x|D,\theta) = -E(f(x)|D,\theta)$$

Acquisition function: a feature of a distribution

- **Minimum mean**:

$$a(x|D,\theta) = -E(f(x)|D,\theta) = -\int f(x)p(f|D,\theta)df.$$

...minimum variance also possible.

Acquisition function: a feature of a distribution

- **Minimum mean**:

$$a(x|D,\theta) = -E(f(x)|D,\theta) = -\int f(x)p(f|D,\theta)df.$$

...minimum variance also possible.

- **Probability of improvement** (below the best so far $f(x_b)$):

$$a(x|D,\theta) = \Phi(\gamma(x)), \qquad \text{where} \qquad \gamma(x) = \frac{f(x_b) - E(f(x)|D,\theta)}{\sqrt{Var(f(x)|D,\theta)}}.$$

Acquisition function: a feature of a distribution

- **Minimum mean**:

$$a(x|D,\theta) = -E(f(x)|D,\theta) = -\int f(x)p(f|D,\theta)df.$$

...minimum variance also possible.

- **Probability of improvement** (below the best so far $f(x_b)$):

$$a(x|D,\theta) = \Phi(\gamma(x)), \qquad \text{where} \qquad \gamma(x) = \frac{f(x_b) - E(f(x)|D,\theta)}{\sqrt{Var(f(x)|D,\theta)}}.$$

- **Expected improvement**:

$$a(x|D,\theta) = \sqrt{Var(f(x)|D,\theta)}\Big(\gamma(x)\Phi(\gamma(x)) + \mathcal{N}_{0,1}(\gamma(x))\Big).$$

Acquisition function: a feature of a distribution

- **Minimum mean**:

$$a(x|D,\theta) = -E(f(x)|D,\theta) = -\int f(x)p(f|D,\theta)df.$$

...minimum variance also possible.

- **Probability of improvement** (below the best so far $f(x_b)$):

$$a(x|D,\theta) = \Phi(\gamma(x)), \qquad \text{where} \qquad \gamma(x) = \frac{f(x_b) - E(f(x)|D,\theta)}{\sqrt{Var(f(x)|D,\theta)}}.$$

- **Expected improvement**:

$$a(x|D,\theta) = \sqrt{Var(f(x)|D,\theta)}\Big(\gamma(x)\Phi(\gamma(x)) + \mathcal{N}_{0,1}(\gamma(x))\Big).$$

- **GP lower confidence bound**:

$$a(x|D,\theta) = E(f(x)|D,\theta) - \kappa\sqrt{Var(f(x)|D,\theta)}.$$

Comparing acquisition functions

Comparing acquisition functions

- EI and GP-UCB are most popular.

Comparing acquisition functions

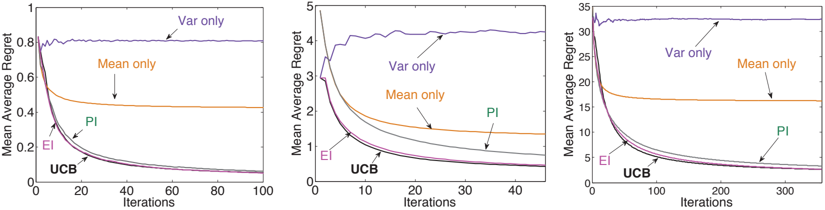
- EI and GP-UCB are most popular.
- Generally EI is empirically preferred, UCB theoretically preferred.

Comparing acquisition functions

- EI and GP-UCB are most popular.
- Generally EI is empirically preferred, UCB theoretically preferred.
- [SKKS09] proved sublinear regret bounds on GP-UCB.

Comparing acquisition functions

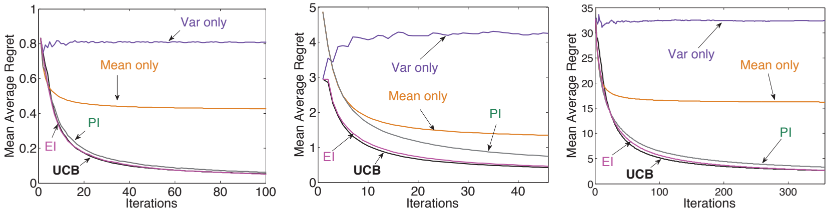
- EI and GP-UCB are most popular.
- Generally EI is empirically preferred, UCB theoretically preferred.
- [SKKS09] proved sublinear regret bounds on GP-UCB.
- Comparing acquisition functions:



(a) *Squared exponential* (b) *Temperature data* (c) *Traffic data*

Comparing acquisition functions

- EI and GP-UCB are most popular.
- Generally EI is empirically preferred, UCB theoretically preferred.
- [SKKS09] proved sublinear regret bounds on GP-UCB.
- Comparing acquisition functions:



(a) *Squared exponential* (b) *Temperature data* (c) *Traffic data*

- Still others exist (e.g. entropy search [HS12]).

GP hyperparameters matter

GP hyperparameters matter

- [SLA12] advocates integrating out gp hyperparameters:

$$\hat{a}(x|D) = \int a(x|D,\theta)p(\theta)d\theta.$$

GP hyperparameters matter

- [SLA12] advocates integrating out gp hyperparameters:

$$\hat{a}(x|D) = \int a(x|D, \theta) p(\theta) d\theta.$$

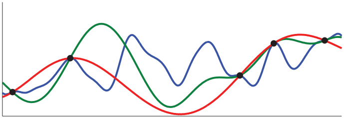
- Consider the effect that a different θ can have on EI...

GP hyperparameters matter

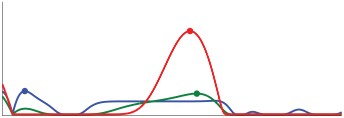
- [SLA12] advocates integrating out gp hyperparameters:

$$\hat{a}(x|D) = \int a(x|D,\theta)p(\theta)d\theta.$$

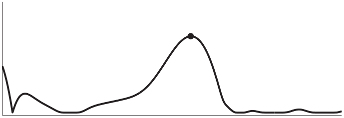
- Consider the effect that a different θ can have on EI...



(a) Posterior samples under varying hyperparameters



(b) Expected improvement under varying hyperparameters



(c) Integrated expected improvement

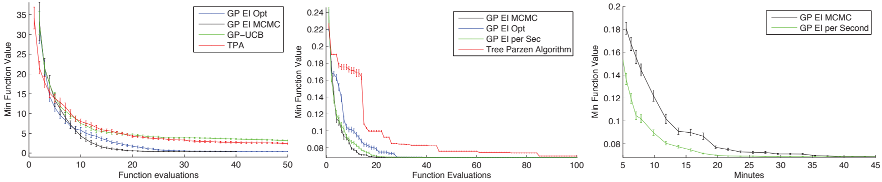
BO on Branin-Hoo and MNIST

BO on Branin-Hoo and MNIST

- Branin-Hoo (left) is a standard test function with known global optima.

BO on Branin-Hoo and MNIST

- Branin-Hoo (left) is a standard test function with known global optima.
- MNIST (middle, right) is a digit classification set.



BO on a three layer convnet

BO on a three layer convnet

- Neural networks are notoriously parameter sensitive...

BO on a three layer convnet

- Neural networks are notoriously parameter sensitive...
- ...and costly to train and evaluate.

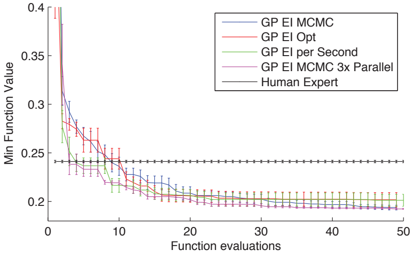
BO on a three layer convnet

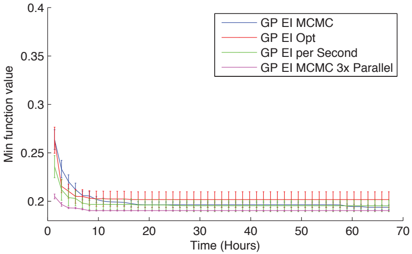
- Neural networks are notoriously parameter sensitive...
- ...and costly to train and evaluate.
- This convnet has 9 hyperparameters, which [SLA12] use BO to optimize.

BO on a three layer convnet

- Neural networks are notoriously parameter sensitive...
- ...and costly to train and evaluate.
- This convnet has 9 hyperparameters, which [SLA12] use BO to optimize.

  ...learning rate, four weight costs, parameters of the response function, and number of epochs (?)





BO summary

BO summary

- Model an expensive-to-evaluate function as a gp.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

...though this is a bit of cheating, since there is a grid.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:
  - constrained optimization

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:
  - constrained optimization
  - high-dimensional optimization

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:
  - constrained optimization
  - high-dimensional optimization
  - using deepnets instead of gp

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:
  - constrained optimization
  - high-dimensional optimization
  - using deepnets instead of gp
  - etc.

BO summary

- Model an expensive-to-evaluate function as a gp.
- Use the gp to make educated guesses where the optimum is.
- Sometimes called *surrogate global optimization*.
- Note global feature: no gradients, so really not a local search method.

  ...though this is a bit of cheating, since there is a grid.
- BO works well and has a rapidly growing literature, for things like:
  - constrained optimization
  - high-dimensional optimization
  - using deepnets instead of gp
  - etc.
- Some doubts remain... e.g., is BO a toy solution?

Outline

Learning the entire function

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

...active learning is a redundant term for optimal experimental design.

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

  ...active learning is a redundant term for optimal experimental design.
- Standard greedy choice is maximally to reduce posterior entropy of f:

$$\arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right)$$

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

  ...active learning is a redundant term for optimal experimental design.

- Standard greedy choice is maximally to reduce posterior entropy of f:

$$\begin{aligned} & \arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right) \\ = \quad & \arg\max_x I(f;y|x,D) \end{aligned}$$

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

  ...active learning is a redundant term for optimal experimental design.
- Standard greedy choice is maximally to reduce posterior entropy of f:

$$\begin{aligned} & \arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right) \\ = \quad & \arg\max_x I(f;y|x,D) \\ = \quad & \arg\max_x I(y;f|x,D) \end{aligned}$$

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

...active learning is a redundant term for optimal experimental design.

- Standard greedy choice is maximally to reduce posterior entropy of f:

$$
\begin{aligned}
& \arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right) \\
= \quad & \arg\max_x I(f;y|x,D) \\
= \quad & \arg\max_x I(y;f|x,D) \\
= \quad & \arg\max_x H(y|x,D) - E_{f|D}\left(H(y|x,f)\right)
\end{aligned}
$$

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

  ...active learning is a redundant term for optimal experimental design.
- Standard greedy choice is maximally to reduce posterior entropy of f:

$$
\begin{aligned}
& \arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right) \\
= \quad & \arg\max_x I(f;y|x,D) \\
= \quad & \arg\max_x I(y;f|x,D) \\
= \quad & \arg\max_x H(y|x,D) - E_{f|D}\left(H(y|x,f)\right) \\
= \quad & \arg\max_x H(E_{f|D}(y|x,f)) - E_{f|D}\left(H(y|x,f)\right).
\end{aligned}
$$

Learning the entire function

- BO uses a gp surrogate to find the optima of an expensive function.
- Bayesian active learning uses a gp surrogate to learn the expensive function.

  ...active learning is a redundant term for optimal experimental design.

- Standard greedy choice is maximally to reduce posterior entropy of f:

$$
\begin{aligned}
& \arg\max_x H(f|D) - E_{y|D}\left(H(f|y,x,D)\right) \\
= \quad & \arg\max_x I(f;y|x,D) \\
= \quad & \arg\max_x I(y;f|x,D) \\
= \quad & \arg\max_x H(y|x,D) - E_{f|D}\left(H(y|x,f)\right) \\
= \quad & \arg\max_x H(E_{f|D}(y|x,f)) - E_{f|D}\left(H(y|x,f)\right).
\end{aligned}
$$

- The point x that maximally reduces the uncertainty (entropy) in f (namely $H(f|D)$, down to $E_y\left(H(f|y,x,D)\right)$, the expected result) is the point x has maximal mutual information between $f(x|D)$ and the noisy observation y.

Active learning in a gp classification setting

Active learning in a gp classification setting

- Recall gp classification:

$$f \sim \mathcal{GP}(0, k), \qquad \text{and} \qquad y_i | f(x_i) \sim Bern\big(\Phi(f(x_i))\big).$$

Active learning in a gp classification setting

- Recall gp classification:

$$f \sim \mathcal{GP}(0, k), \qquad \text{and} \qquad y_i | f(x_i) \sim Bern\big(\Phi(f(x_i))\big).$$

- Our greedy choice then operates on the Bernoulli entropy:

$$h(p) = -p \log p - (1-p) \log(1-p).$$

...resulting in the greedy objective function:

$$I(f; y | x, D) = h(E_{f|D}(\Phi(f(x)))) - E_{f|D}(h(\Phi(f(x)))),$$

...which is intractable but only one dimensional, hence quickly solved.

Active learning in a gp classification setting

- Recall gp classification:

$$f \sim \mathcal{GP}(0, k), \qquad \text{and} \qquad y_i | f(x_i) \sim Bern\big(\Phi(f(x_i))\big).$$

- Our greedy choice then operates on the Bernoulli entropy:

$$h(p) = -p \log p - (1-p)\log(1-p).$$

  ...resulting in the greedy objective function:

$$I(f; y|x, D) = h(E_{f|D}(\Phi(f(x)))) - E_{f|D}(h(\Phi(f(x)))),$$

  ...which is intractable but only one dimensional, hence quickly solved.

- Maximize this information gain at each step $\rightarrow$ active learning.

Application: hearing tests [GSW$^+$15]

Application: hearing tests [GSW$^+$15]

- The doctor plays tones at different frequencies and amplitudes.

Application: hearing tests [GSW$^+$15]

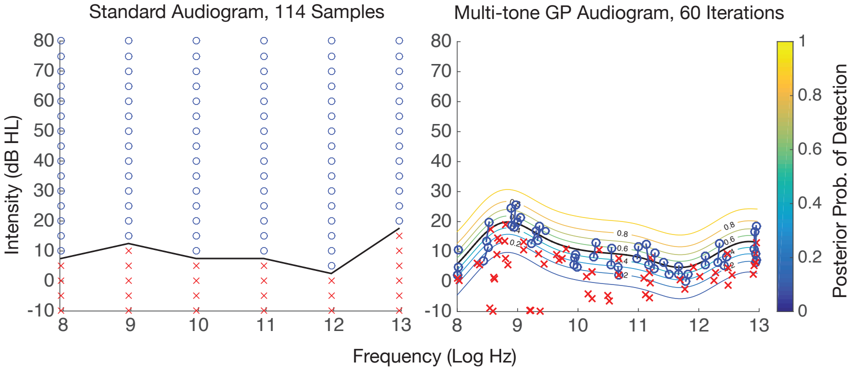
- The doctor plays tones at different frequencies and amplitudes.
- The patient gives a binary report (heard or not heard).

Application: hearing tests [GSW+15]

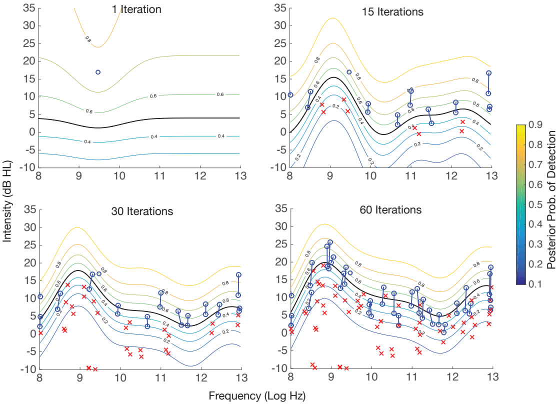
- The doctor plays tones at different frequencies and amplitudes.
- The patient gives a binary report (heard or not heard).
- The object of interest is the *audiogram*, a discriminability function.

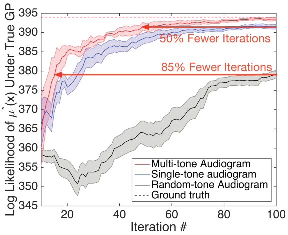
Application: hearing tests [GSW$^+$15]

- The doctor plays tones at different frequencies and amplitudes.
- The patient gives a binary report (heard or not heard).
- The object of interest is the *audiogram*, a discriminability function.
- Note: [GSW$^+$15] extends to multiple simultaneous tones.



Application: hearing tests [GSW+15]





Outline

References

[GSW+15] Jacob R Gardner, Xinyu Song, Kilian Q Weinberger, Dennis Barbour, and John P Cunningham.
Psychophysical detection testing with bayesian active learning.
UAI, 2015.

[HHGL11] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel.
Bayesian active learning for classification and preference learning.
arXiv preprint arXiv:1112.5745, 2011.

[HS12] Philipp Hennig and Christian J Schuler.
Entropy search for information-efficient global optimization.
The Journal of Machine Learning Research, 13(1):1809–1837, 2012.

[Jon01] Donald R Jones.
A taxonomy of global optimization methods based on response surfaces.
Journal of global optimization, 21(4):345–383, 2001.

[MTZ78] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas.
The application of bayesian methods for seeking the extremum.
Towards Global Optimization, 2(117-129):2, 1978.

[SKKS09] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger.
Gaussian process optimization in the bandit setting: No regret and experimental design.
arXiv preprint arXiv:0912.3995, 2009.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P Adams.
Practical bayesian optimization of machine learning algorithms.
In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[Wah90] Grace Wahba.
Spline models for observational data, volume 59.
Siam, 1990.