STAT G8325
Gaussian Processes and Kernel Methods
Lecture Notes §02: Model Selection

John P. Cunningham

Department of Statistics
Columbia University

Outline

Recap of §01

Administrative interlude

The problem of model selection

Hyperparameter optimization for gp model selection

Cross-validation for gp model selection

Sampling for gp model selection (with review)

Slice sampling for gp model selection [MA10]

Approximate integration for gp model selection [GOH14, §3]

Outline

Gaussian process

Definition (gaussian process). Let:

i. T *be any set,*

ii. $m : T \to \mathbb{R}$ *be any function, and*

iii. $k : T \times T \to \mathbb{R}$ *be any function that is symmetric (i.e.,* $k(t, t') = k(t', t) \forall t, t' \in T$*) and positive definite (i.e., for any finite* $G \subset T$*, the matrix* $K_G = \{k(t, t')\}_{t,t' \in G}$ *is positive definite).*

Then there exists a gaussian process $f = \{f_t\}_{t \in T}$ *with mean function* m *and covariance* k*. We write* $f \sim \mathcal{GP}(m, k)$*. Notably,* $f \sim \mathcal{GP}(m, k)$ *if and only if, for every finite* $G \subset T$*, the consistent collection of random variables* $f_G \sim \mathcal{N}(m_G, K_G)$*.*

- The induced finite marginals are $\mathcal{N}(m_G, K_G)$, with:

$$m_G = \begin{bmatrix} m(t_1) \\ \vdots \\ m(t_{|G|}) \end{bmatrix} \in \mathbb{R}^{|G|}, \quad K_G = \{k(t, t')\}_{t,t' \in G}, \quad \forall G \subset T.$$

Review: multivariate gaussian

- $f \in \mathbb{R}^n$ is normally distributed means:

$$p(f) = (2\pi)^{-\frac{n}{2}} |K|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(f-m)^\top K^{-1}(f-m)\right\}$$

  for mean vector $m \in \mathbb{R}^n$ and covariance matrix $K \in \mathbb{R}^{n \times n}$.

- shorthand: $f \sim \mathcal{N}(m, K)$

- Again, $f \sim \mathcal{GP}(m, k)$ is a Gaussian process if $f(t) = [f(t_1), ..., f(t_n)]'$ has a consistent multivariate normal distribution for all choices of $n \in \mathbb{N}$ and $t = [t_1, ..., t_n]'$:

$$f(t) \sim \mathcal{N}(m(t), k(t,t)).$$

GP regression basics

- n input points $t \in \mathbb{R}^n$ (reminder: implied input domain $\mathbb{R}$)
- prior (or latent) $f \sim \mathcal{GP}(m_f, k_{ff})$
- additive iid noise $\epsilon \sim \mathcal{GP}(0, \sigma_\epsilon^2 \delta)$
- let $y = f + \epsilon$, then (using additivity of GP):

$$p(y(t), f(t)) = p(y|f)p(f) = \mathcal{N}\left(\begin{bmatrix} f \\ y \end{bmatrix}; \begin{bmatrix} m_f \\ m_y \end{bmatrix}, \begin{bmatrix} K_{ff} & K_{fy} \\ K_{fy}^T & K_{yy} \end{bmatrix}\right)$$

- Regression/inference/conditioning:

$$f|y \sim \mathcal{N}\left(m_f + K_{fy}K_{yy}^{-1}(y - m_y) \quad , \quad K_{ff} - K_{fy}K_{yy}^{-1}K_{yf}\right)$$

- Prediction at $y^* = y(t^*)$:

$$y^*|y \sim \mathcal{N}\left(m_{y^*} + K_{y^*y}K_{yy}^{-1}(y - m_y) \quad , \quad K_{y^*y^*} - K_{y^*y}K_{yy}^{-1}K_{yy^*}\right)$$

Marginalization (marginal likelihood and model selection)

- Again, if:

$$\begin{bmatrix} f \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_f \\ m_y \end{bmatrix}, \begin{bmatrix} K_{ff} & K_{fy} \\ K_{yf} & K_{yy} \end{bmatrix} \right)$$

- we can marginalize out the latent:

$$p(y) = \int p(y|f)p(f)df \qquad \leftrightarrow \qquad y \sim \mathcal{N}(m_y, K_{yy})$$

- $\log(p(y))$ is the data log-likelihood of the data (aka marginal likelihood)
- GP have hyperparameters θ which influence $\log(p(y))$
- Dealing with that fact is the subject of model selection...

Outline

Useful information

- Always start with the syllabus. Highlights...
- Prerequisites (aka, did §01 make sense to you):
  - Stochastic processes to a basic understanding of gaussian processes
  - Machine learning such as W4400
  - Probability, statistics, linear algebra, basic convex optimization
  - Programming skills
- Grade:
  - **Homework (10%).** Two or three homework sets will be given to ensure students are keeping pace. Homework will contain both written and programming/data analysis elements.
  - **Attendance and Participation (40%).** The course will have a seminar format, and your involvement is critical. This means read in advance, and demonstrate that knowledge.
  - **Course Project (50%).** The course projects will be the focus of the latter half of this course. Projects can take a variety of forms, from contributing to open source machine learning projects, to analyzing data of interest, to advancing a theoretical topic. We will spend substantial time developing ideas for projects, tracking and discussing progress, and presenting final work product. Individual projects are ideal, though projects with groups of two may also be appropriate.

Progress...

Week	Content
1	Introduction to gaussian processes for machine learning • Reading: [RW06, ch. 1-2] • HW1 out: `https://github.com/cunni/gpkm/blob/master/hw1.ipynb`
2	Model selection • Reading: [RW06, ch. 5.1-5.4]; [MA10]; [GOH14, §3 only] • HW1 ongoing
3	Approximate inference • Reading: [KR05]; [RMC09]; [RW06, ch. 3; 5.5]; [HMG15] • HW1 due at the beginning of Monday lecture

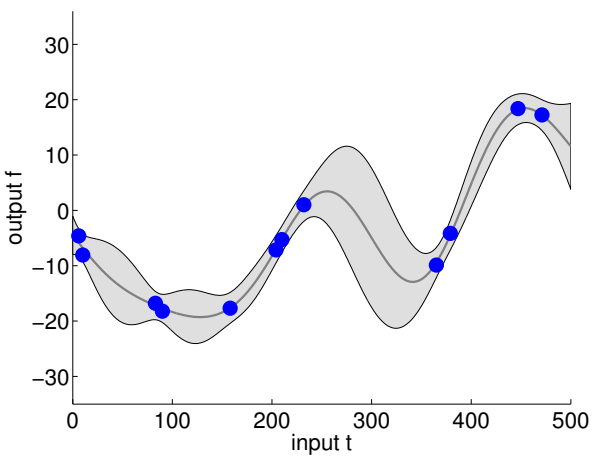
- Note: §02 could take a day or two weeks... let's discuss.

Outline

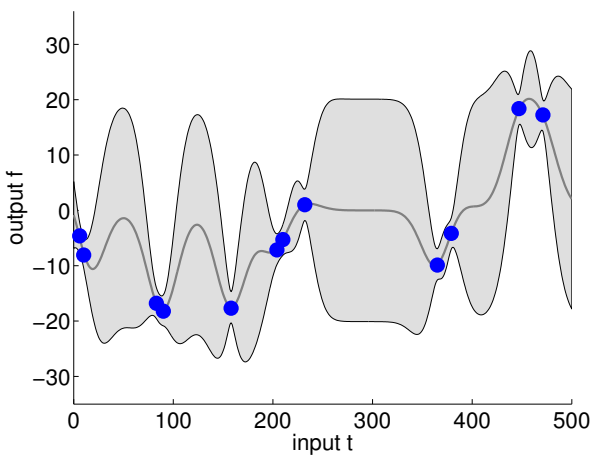
Model selection / hyperparameter learning

- $f \sim \mathcal{GP}(0, k_{ff})$, where $k_{ff}(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\}$
- $\epsilon \sim \mathcal{GP}(0, \sigma_\epsilon^2 \delta)$, where $\delta(t_i, t_j) = \mathbb{I}(t_i = t_j)$.
- $y = f + \epsilon$ is observed data:

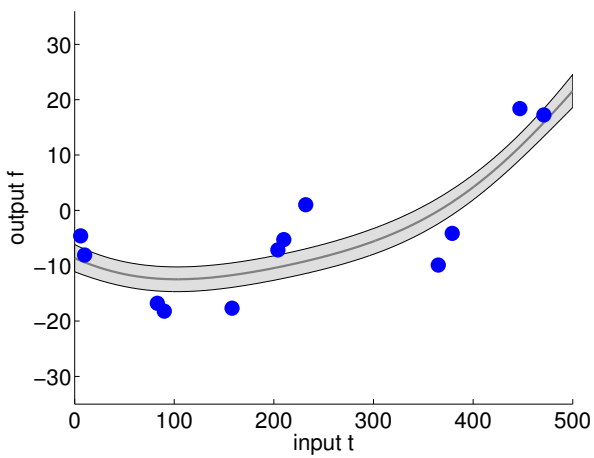
$\ell = 50$: just right



$\ell = 15$: overfitting



$\ell = 250$: underfitting



Occam's razor in probabilistic machine learning

- Recall a *model* $\mathcal{H} = \{P_\theta : \theta \in \Theta\}$: a family of probability distributions indexed over some parameter space Θ.
- Example:

$$\begin{aligned}
\mathcal{H}_1 &= \left\{ \mathcal{GP}(0, k_\theta) : k_\theta = \sigma_f^2 \exp\left\{ -\frac{1}{2\ell^2}(t_i - t_j)^2 \right\} \text{ is squared exponential with } \theta = \{\sigma_f, \ell\} \right\} \\
\mathcal{H}_2 &= \{ \mathcal{GP}(0, k_\theta) : k_\theta = \theta \min(t_i, t_j) \text{ is scaled brownian motion} \} .
\end{aligned}$$

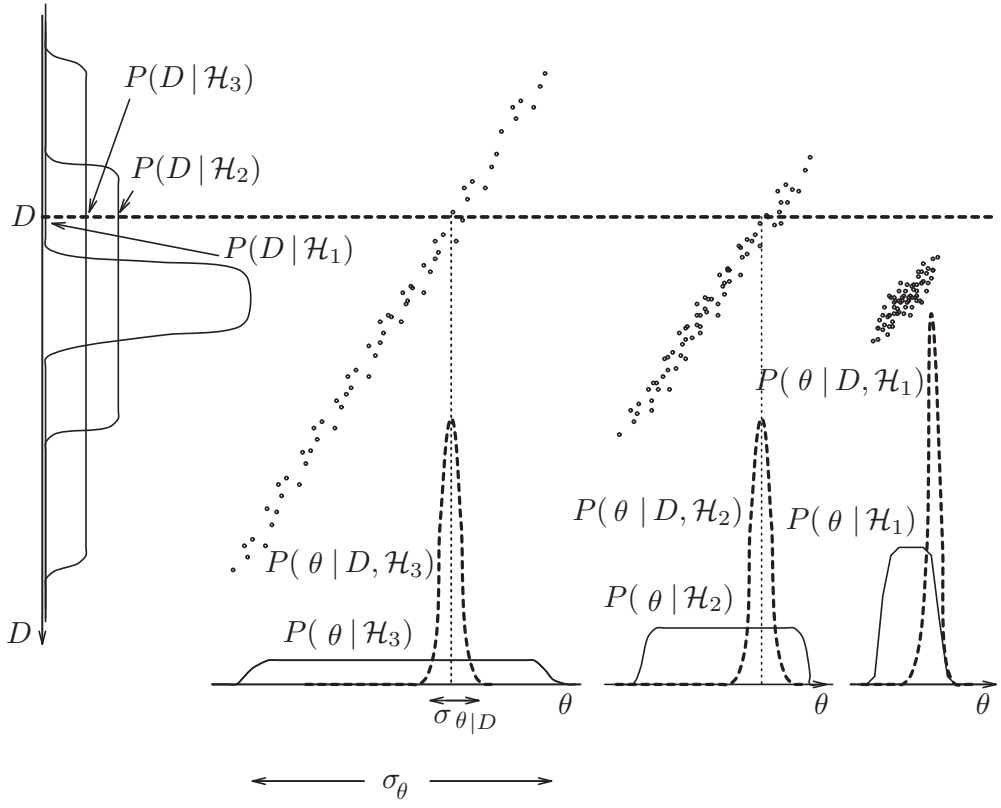
- With $p(\mathcal{H}_1) = p(\mathcal{H}_2)$ and data D, we want to *select a model* by comparing:

$$p(H_i|D) \propto P(D|H_i) = \int_\Theta P(D|\theta, H_i) P(\theta|H_i) d\theta \quad \forall \;\; i \in \{1, 2\}.$$

- Note *model selection* is often used in the sense of selecting a particular P_θ from a model $\mathcal{H}$. We will also sometimes use this (improper?) convention.

Occam's razor in probabilistic machine learning

- "Plurality should not be assumed without necessity"
- Example: data is a **single point** $D = \theta + \epsilon$, for iid noise ϵ and parameter θ.



- Probabilistic ML can balance complexity and avoid overfitting naturally.
- See [Mac03, ch. 28] for more (or [RW06, ch. 5]).

Occam's razor in gp

- $y \sim GP(m, k_\theta)$ with $k_\theta = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\} + \sigma_\epsilon^2 \delta(t_i - t_j)$:

$$\log\left(p(y|\theta)\right) = -\frac{1}{2}(y-m)^\top K_\theta^{-1}(y-m) - \frac{1}{2}\log|K_\theta| - \frac{n}{2}\log(2\pi).$$

- Read this as: *data fit term + complexity penalty + constant*.
- First get comfortable with the tradeoff in σ_ϵ and σ_f.
- Next consider ℓ:
  - $-\frac{1}{2}\log|K_\theta|$ increases in ℓ

    ...determinant as volume of K_θ; or $\sigma_\epsilon = 0, \ell \to \infty \Rightarrow \lambda_n(K_\theta) \to 0$
  - $-\frac{1}{2}(y-m)^\top K_\theta^{-1}(y-m)$ decreases in ℓ

    ...less flexibility to be distant from the mean
- GP complexity increases with decreasing ℓ and increasing $\sigma_f, \sigma_\epsilon$.

Model likelihood

- $y \sim GP(m, k_\theta)$ with $k_\theta = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\} + \sigma_\epsilon^2 \delta(t_i - t_j)$:

$$\log\left(p(y|\theta)\right) = -\frac{1}{2}(y-m)^\top K_\theta^{-1}(y-m) - \frac{1}{2}\log|K_\theta| - \frac{n}{2}\log(2\pi).$$

- Our desired quantity is then the likelihood of the data under the model:

$$p(H|D) \propto P(D|H) = \int_\Theta P(D|\theta, H)P(\theta|H)d\theta.$$

- How can we deal with this intractable integral?
  - $\approx$ point estimate ("ML-II optimization"; [RW06, ch. 5.1-5.4])
  - $\approx$ sum of samples (sampling methods, MCMC; [MA10])
  - $\approx$ simpler integral (laplace-type integral; [GOH14, §3])
  - $\approx$ a different simpler integral (variational inference; later in the course)

Outline

Model selection (1): marginal likelihood

- We approximate:

$$
\begin{aligned}
p(H|D) \propto P(D|H) &= \int_{\Theta} P(D|\theta, H) P(\theta|H) d\theta \\
&\approx P(D|\theta_{MAP}, H) P(\theta_{MAP}|H) \\
&\text{or} \\
&\approx P(D|\theta_{ML}, H).
\end{aligned}
$$

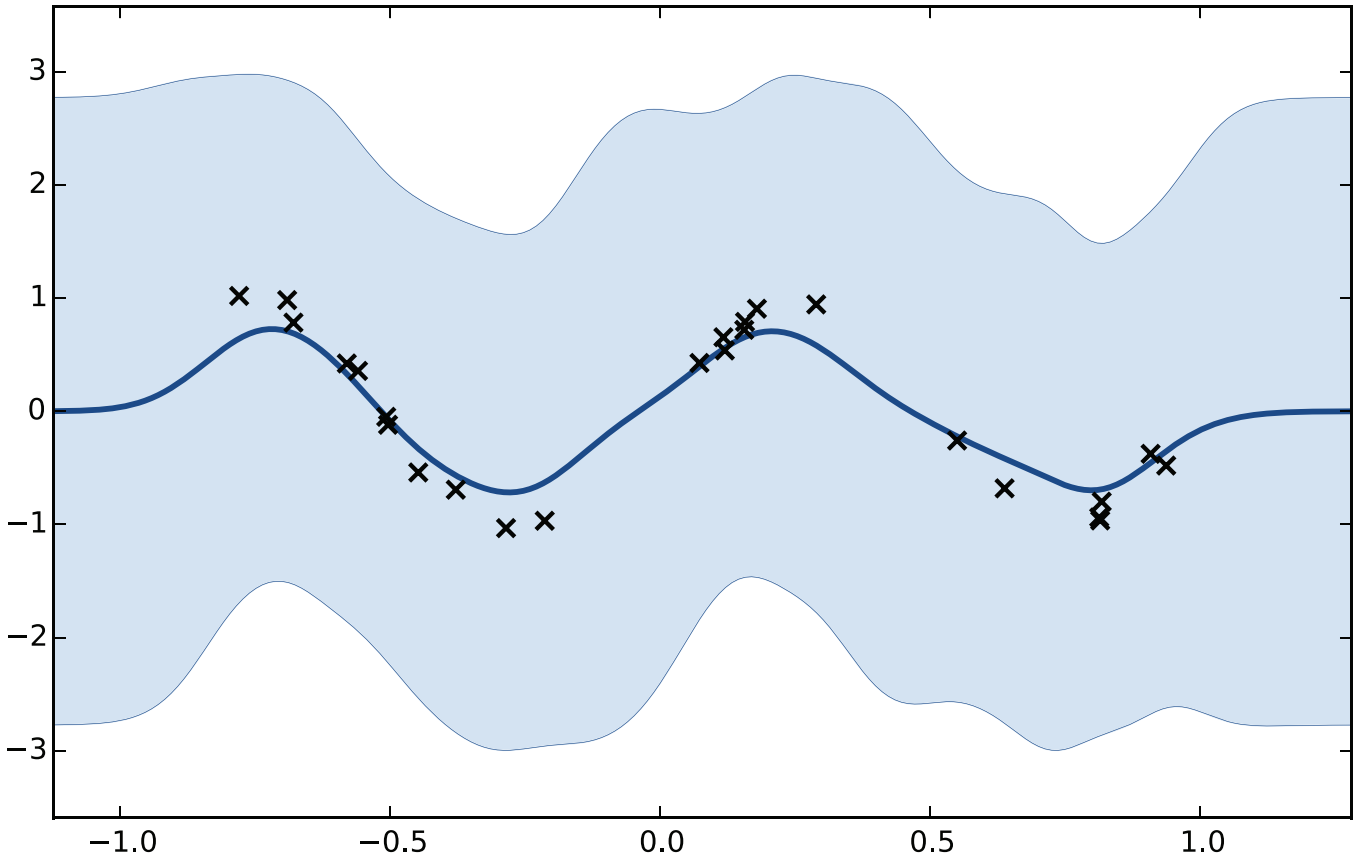
- where θ_{MAP} and θ_{ML} are the corresponding maxima.
- $P(D|\theta_{ML}, H)$ is most common and is colloquially called *ML type II*.
- most tractable, but ignores randomness in θ (perils such as [vdVvZ09]).

ML-II in pictures (hw1)

- We approximate:

$$p(H|D) \quad \approx \quad P(D|\theta_{ML}, H).$$

- Before:

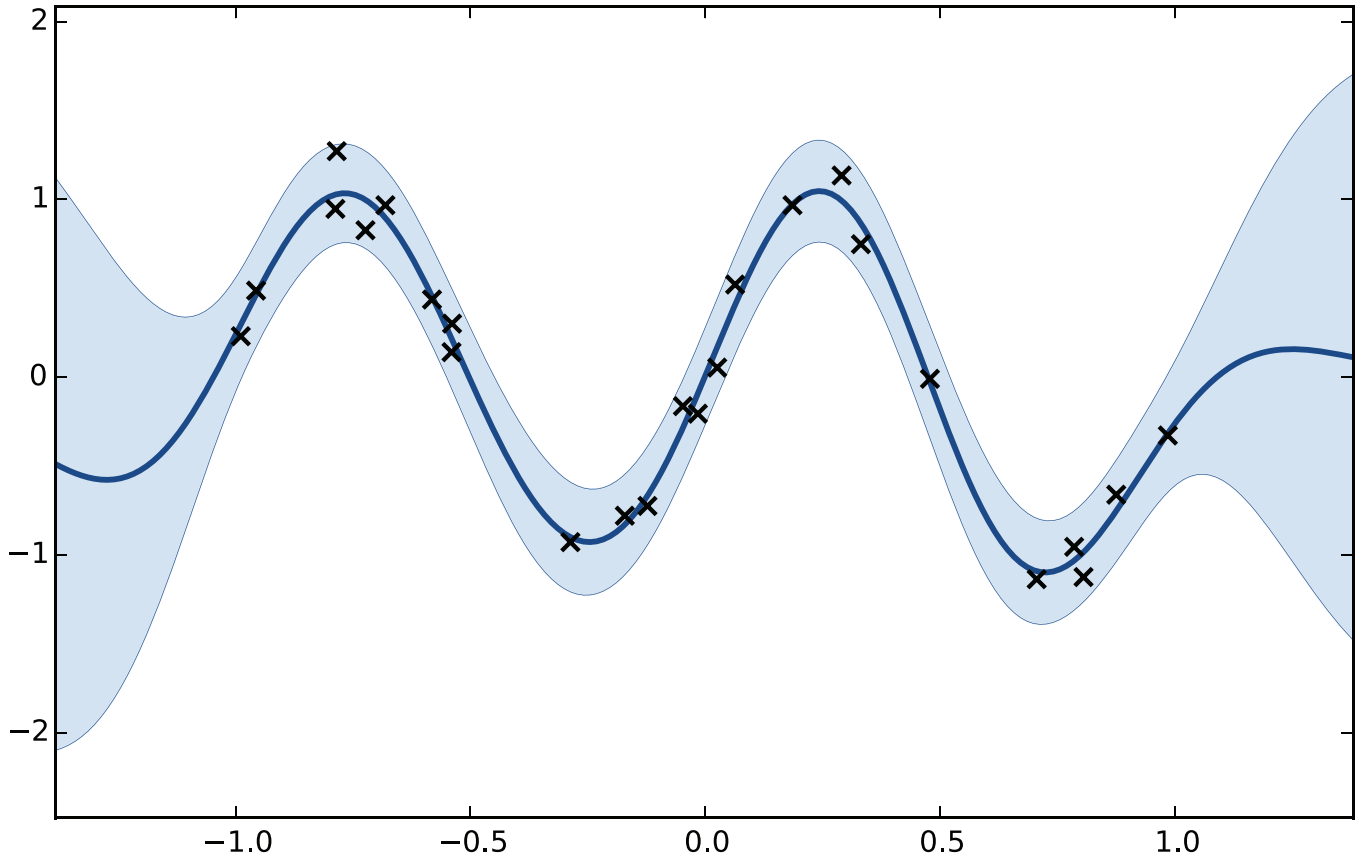


ML-II in pictures (hw1)

- We approximate:

$$p(H|D) \quad \approx \quad P(D|\theta_{ML}, H).$$

- After:



ML-II in equations

- We approximate:

$$\begin{aligned} \arg\max_{\theta} P(D|\theta, H) &= \\ \arg\max_{\theta} \log\left(p(y|\theta)\right) &= -\frac{1}{2} y^\top K_\theta^{-1} y - \frac{1}{2}\log|K_\theta| - \frac{n}{2}\log(2\pi). \end{aligned}$$

- Run your favorite non-convex optimization on $\nabla_\theta \log\left(p(y|\theta)\right)$, with:

$$\begin{aligned} \frac{\partial}{\partial\theta_i}\log\left(p(y|\theta)\right) &= \frac{1}{2} y^\top K_\theta^{-1}\frac{\partial K_\theta}{\partial\theta_i}K_\theta^{-1} y - \frac{1}{2}\mathsf{tr}\left(K^{-1}\frac{\partial K_\theta}{\partial\theta_i}\right) \\ &= \frac{1}{2}\left(\left(\alpha\alpha^\top - K^{-1}\right)\frac{\partial K_\theta}{\partial\theta_i}\right), \quad \text{with} \quad \alpha = K_\theta^{-1} y. \end{aligned}$$

- central to all GP implementations (`optimize()` in `model.py`, `minimize.m` in `gpml`, ...).

Outline

Model selection (2): cross validation

- Can also consider *leave-one-out* predictive likelihood to penalize overfitting:

$$\begin{aligned}\mathcal{L}_{LOOCV}(\theta) &= \sum_{i=1}^{n} \log p(y_i|y_{-i},\theta) \\ &\propto -\frac{1}{2}\log\sigma_i^2 - \frac{1}{2\sigma_i^2}(y_i-\mu_i)^2.\end{aligned}$$

- Fiddling with the Schur trick again, we get:

$$\mu_i = y_i - \frac{\left\{K^{-1}y\right\}_i}{\left\{K^{-1}\right\}_{ii}}, \qquad \sigma_i^2 = \frac{1}{\left\{K^{-1}\right\}_{ii}}.$$

- Convince yourself μ_i is independent of y_i.
- Some further tricks make this not terribly burdensome, but still a factor of $|\theta|$ larger than ML-II. See [VTMW14] for new directions in this topic.

Outline

Model selection (3): sampling

- We might try vanilla Monte Carlo:

$$\begin{aligned} p(H|D) \propto P(D|H) &= \int_{\Theta} P(D|\theta, H) P(\theta|H) d\theta \\ &\approx \frac{1}{m} \sum_{j=1}^{m} P(D|\theta_j, H) \quad \text{where} \quad \theta_j \sim P(\theta|H) \end{aligned}$$

- However, if Θ is even reasonably big:

$$k_\theta = \sigma_f^2 \exp\left\{ -\sum_{d=1}^{D} \frac{1}{2\ell_d^2} (t_i^d - t_j^d)^2 \right\} + \sigma_\epsilon^2 \delta(t_i - t_j) \quad \text{i.e.} \quad \Theta = \mathbb{R}^{D+1}$$

- (pause for a moment to think through the ARD properties of this kernel)...
- Then this estimator has huge variance $\rightarrow$ bad estimate of $P(D|H)$

why? impossible to meaningfully cover Θ

Importance sampling as motivation for MCMC

- Next we might try importance sampling:

$$Z \triangleq P(D) = \int_\Theta \frac{P(D|\theta)P(\theta)}{Q(\theta)} Q(\theta) d\theta \quad \approx \quad \frac{1}{m} \sum_{j=1}^{m} \frac{P(D|\theta_j)P(\theta_j)}{Q(\theta_j)} \qquad \text{where} \quad \theta_j \sim Q(\theta).$$

- Also terrible: estimator will be dominated by a few samples.
- But notice that won't be the case if $Q(\theta)$ is close to $\propto P(D|\theta)P(\theta)$.

offhand this comment seems vacuous... why?

- Great trick: calculate instead a ratio of normalizers of "close" distributions:

$$\begin{aligned} \frac{Z_{\alpha_k}}{Z_{\alpha_{k-1}}} &= \int \frac{P(D|\theta)^{\alpha_k} P(\theta)}{P(D|\theta)^{\alpha_{k-1}} P(\theta)} \left(\frac{1}{Z_{\alpha_{k-1}}} P(D|\theta)^{\alpha_{k-1}} P(\theta) \right) d\theta \\ &\approx \frac{1}{m} \sum_{j=1}^{m} \frac{P(D|\theta_j)^{\alpha_k} P(\theta_j)}{P(D|\theta_j)^{\alpha_{k-1}} P(\theta_j)}. \end{aligned}$$

proof of ratio? $1 = \int p = \int \frac{p}{q} q \ldots$

- This importance sampler will work brilliantly if α_k is close to α_{k-1}...

Annealed importance sampling

- If we can get:

$$\frac{Z_{\alpha_k}}{Z_{\alpha_{k-1}}} \approx \frac{1}{m}\sum_{j=1}^{m}\frac{P(D|\theta_j)^{\alpha_k}P(\theta_j)}{P(D|\theta_j)^{\alpha_{k-1}}P(\theta_j)}.$$

- Make a schedule $\alpha_0 = 0 < ... < \alpha_K = 1$. Then:

$$\frac{Z_{\alpha_K}}{Z_{\alpha_0}} = \frac{Z_{\alpha_K}}{Z_{\alpha_{K-1}}}\cdot\frac{Z_{\alpha_{K-1}}}{Z_{\alpha_{K-2}}}\cdot ... \cdot \frac{Z_{\alpha_k}}{Z_{\alpha_{k-1}}}\cdot ... \cdot \frac{Z_{\alpha_1}}{Z_{\alpha_0}} = \frac{P(D)}{Z_0}.$$

- Since Z_0 is the known normalizer of the prior $P(\theta)$, we have $Z_1 = P(D)$!

- AIS is the gold-standard method for calculating normalizing constants.

- What is missing?

How do we sample from $Q_{k-1}(\theta) \propto P(D|\theta)^{\alpha_{k-1}}P(\theta)$?

Core idea of MCMC

- Idea: wander around Θ, biasing ourselves to higher $P(\theta|D)$ regions.

  cf. lack of memory in importance/prior/rejection sampling.

- Recall: Markov chain
  - A sufficiently nice Markov chain has an invariant distribution π_{inv}.

    sufficiently nice? aperiodic and irreducible
  - At convergence, each sample θ_i from the chain has marginal π_{inv}.

- *Definition (Markov chain Monte Carlo).* With a goal to sample from p, define a MC with $\pi_{\text{inv}} \equiv p$. Sample $\theta_1, \theta_2, \ldots$ from the chain. Once the chain has converged, $\theta_i \sim p$.

- Note: θ_{i+1} typically depends on θ_i in a Markov chain, so MCMC "remembers" and remains in regions of high probability.

Continuous markov chains

- Typical Markov chains have a finite state space. For MCMC, state space must be the domain of $P(\theta|D) \rightarrow$ often continuous.
- Continuous Markov chain: initial distribution π_{init} and conditional probability $t(\theta'|\theta)$, the *transition probability* or *transition kernel*.

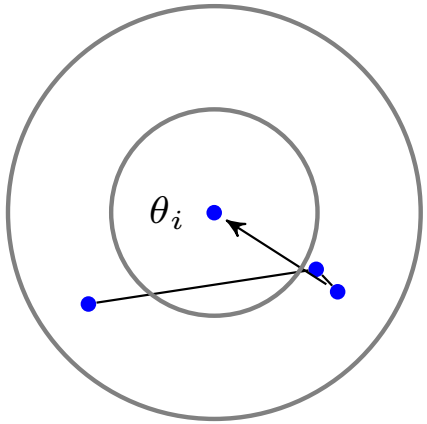
Discrete case: $t(\theta' = i|\theta = j)$ is the entry T_{ij} of the transition matrix T.

- Example Markov chain on $\mathbb{R}^2$:
  - Define a (very) simple Markov chain by sampling

$$\theta_{i+1} \sim \mathcal{N}(\,.\,|\theta_i, \sigma^2)$$

  - In other words, the transition distribution is

$$t(\theta_{i+1}|\theta_i) := \mathcal{N}(\theta_{i+1}|\theta_i, \sigma^2)\ .$$



Gaussian (gray contours) is placed around the current point θ_i to sample θ_{i+1}.

Invariant distribution

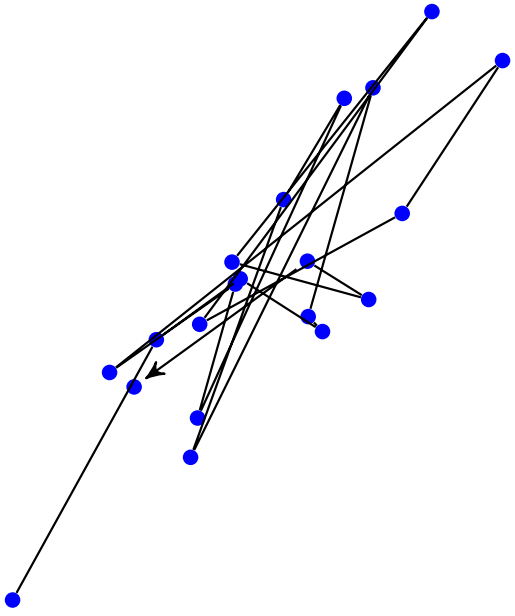
- Recall: Finite case
  - π_{inv} is a distribution on the finite state space Θ.
  - "Invariant" means that, if $\theta_i \sim \pi_{\text{inv}}$, and $\theta_{i+1} \sim t(\,.\,|\theta_i)$ of the chain, then $\theta_{i+1} \sim \pi_{\text{inv}}$.
  - In terms of the transition matrix T, write $T\pi_{\text{inv}} = \pi_{\text{inv}}$.

- Continuous case
  - Θ is now uncountable (e.g. $\Theta = \mathbb{R}^{D+1}$ as in the ARD kernel).
  - The transition matrix $T \rightarrow$ the conditional probability t.
  - A density π_{inv} is invariant if

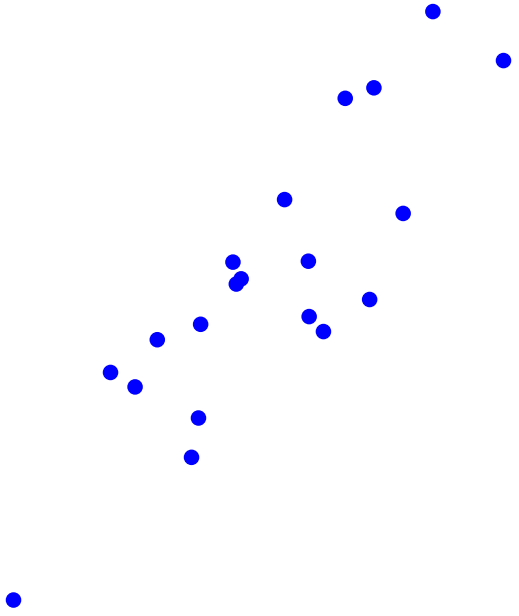
$$\int_{\Theta} t(\theta'|\theta)\pi_{\text{inv}}(\theta)d\theta = \pi_{\text{inv}}(\theta').$$

  - which should feel like the continuous analog of $\sum_i T_{ij}(\pi_{\text{inv}})_i = (\pi_{\text{inv}})_j$.

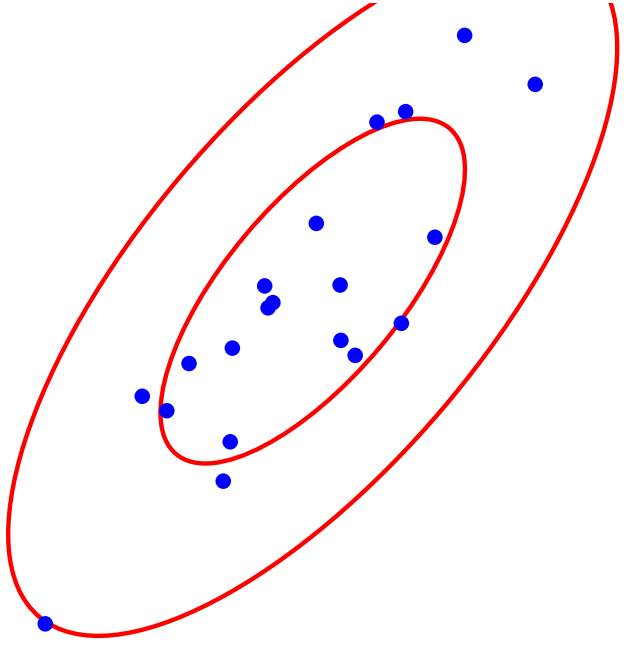
Markov chain sampling



We run the Markov chain n for steps. Each step moves from the current location θ_i to a new θ_{i+1}.



We "forget" the order and regard the locations $\theta_{1:n}$ as a random set of points.



If p (red contours) is both the invariant and initial distribution, each θ_i is distributed as $\theta_i \sim p$.

- Required considerations:
  1. We have to construct a MC with invariant distribution p.
  2. We cannot actually start sampling with $\theta_1 \sim p$; if we knew how to sample from p, all of this would be pointless.
  3. Each point θ_i is *marginally* distributed as $\theta_i \sim p$, but the points are *not* i.i.d.

Consideration 1: Constructing the markov chain

- Reminder: we can evaluate $p(\theta)$ (or $\propto$), but we can't sample...
- Metropolis-Hastings (MH) kernel
  1. We start by defining a conditional probability $q(\theta'|\theta)$ on Θ.

     q has nothing to do with p. We could e.g. choose $q(\theta'|\theta) = \mathcal{N}(\theta'|\theta, \sigma^2)$.
  2. We define the **rejection kernel**

$$A(\theta_{i+1}|\theta_i) := \min\left\{1, \frac{q(\theta_i|\theta_{i+1})p(\theta_{i+1})}{q(\theta_{i+1}|\theta_i)p(\theta_i)}\right\}$$

     Why is knowing $\propto p$ enough here?
  3. Chain transition probability:

$$t(\theta_{i+1}|\theta_i) := q(\theta_{i+1}|\theta_i)A(\theta_{i+1}|\theta_i) + \delta_{\theta_i}(\theta_{i+1})c(\theta_i) \quad \text{where} \quad c(\theta_i) := \int q(\theta'|\theta_i)(1 - A(\theta'|\theta_i))d\theta'$$

     $c(\theta_i)$ is total probability that a proposal is rejected.
- Sampling from the MH chain: At each step $i+1$, generate a proposal $\theta^* \sim q(\,.\,|\theta_i)$ and $U_i \sim \text{Uniform}[0, 1]$.
  - If $U_i \leq A(\theta^*|\theta_i)$, accept proposal: Set $\theta_{i+1} := \theta^*$.
  - If $U_i > A(\theta^*|\theta_i)$, reject proposal: Set $\theta_{i+1} := \theta_i$.

Consideration 2: initial distribution

- Recall: Fundamental theorem on Markov chains Suppose we sample $\theta_1 \sim \pi_{\text{init}}$ and $\theta_{i+1} \sim t(\,.\,|\theta_i)$. This defines a distribution P_i of θ_i, which can change from step to step. If the MC is nice (irreducible and aperiodic), then

$$P_i \to \pi_{\text{inv}} \qquad \text{for} \qquad i \to \infty\,.$$

- Implication:
  - If we can show that $\pi_{\text{inv}} \equiv p$, we do not have to know how to sample from p.
  - Instead, we can start with *any* π_{init}, and will get arbitrarily close to p for sufficiently large i.

Burn-in and mixing time

- The number m of steps required until $P_m \approx \pi_{\text{inv}} \equiv p$ is called the **mixing time** of the Markov chain. (In probability theory, there is a range of definitions for what exactly $P_m \approx \pi_{\text{inv}}$ means.)

- In MC samplers, the first m samples are also called the **burn-in** phase. The first m samples of each run of the sampler are discarded:

$$\underbrace{\theta_1, \ldots, \theta_{m-1}}_{\text{Burn-in; discard.}}, \underbrace{\theta_m, \theta_{m+1}, \ldots}_{\text{Samples from (approximately) } p\text{; keep.}}$$

- In practice, we do not know how large m is. There are a number of methods for assessing whether the sampler has mixed. Such heuristics are often referred to as **convergence diagnostics**.

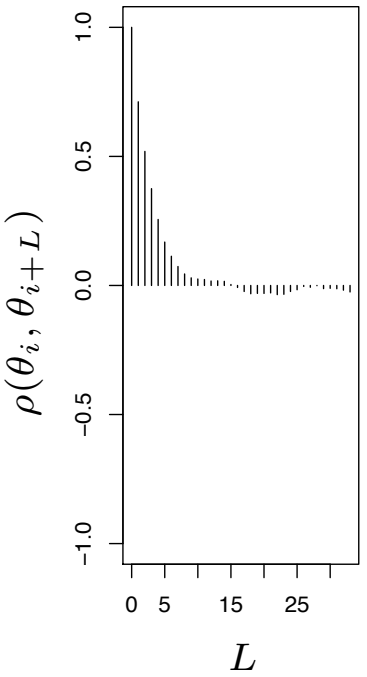
Consideration 3: sequential dependence

- Even after burn-in, the samples from a MC are not iid
- Strategy:
  - Estimate empirically how many steps L are needed for θ_i and θ_{i+L} to be approximately independent. The number L is called the **lag**.
  - After burn-in, keep only every Lth sample; discard samples in between.

- The most common method to estimate lag uses the **autocorrelation function**:

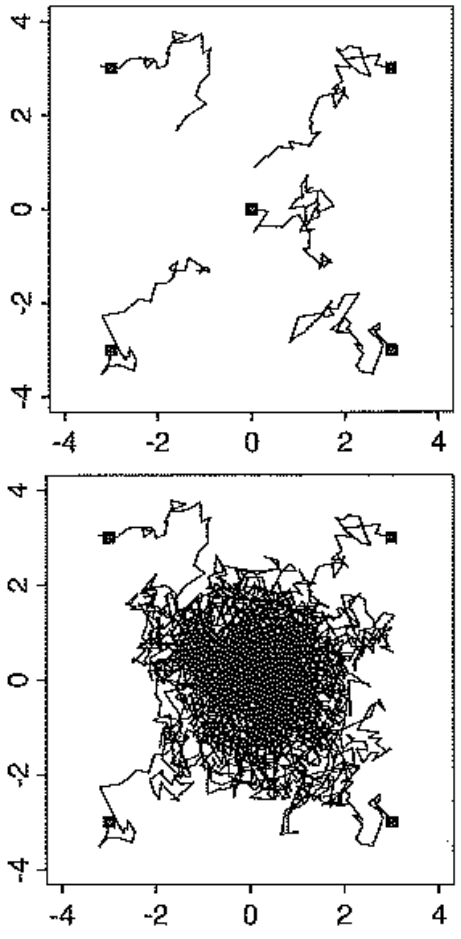
$$\rho(\theta_i, \theta_j) := \frac{\mathbb{E}[\theta_i - \mu_i] \cdot \mathbb{E}[\theta_j - \mu_j]}{\sigma_i \sigma_j}$$

- We compute $\rho(\theta_i, \theta_{i+L})$ empirically from the sample for different values of L, and find the smallest L for which the autocorrelation is close to zero.



Convergence diagnostics

- Gelman-Rubin criterion (popular, not exhaustive)
  - Start several chains at random. For each chain k, sample θ_i^k has a marginal distribution P_i^k.
  - The distributions of P_i^k will differ between chains in early stages.
  - Once the chains have converged, all $P_i = \pi_{\text{inv}}$ are identical.
  - Criterion: Use a hypothesis test to compare P_i^k for different k (e.g. compare P_i^2 against null hypothesis P_i^1). Once the test does not reject anymore, assume that the chains are past burn-in.



MH as stochastic ascent

- The Metropolis-Hastings rejection kernel was defined as:

$$A(\theta_{n+1}|\theta_n) = \min\left\{1, \frac{q(\theta_i|\theta_{i+1})p(\theta_{i+1})}{q(\theta_{i+1}|\theta_i)p(\theta_i)}\right\} .$$

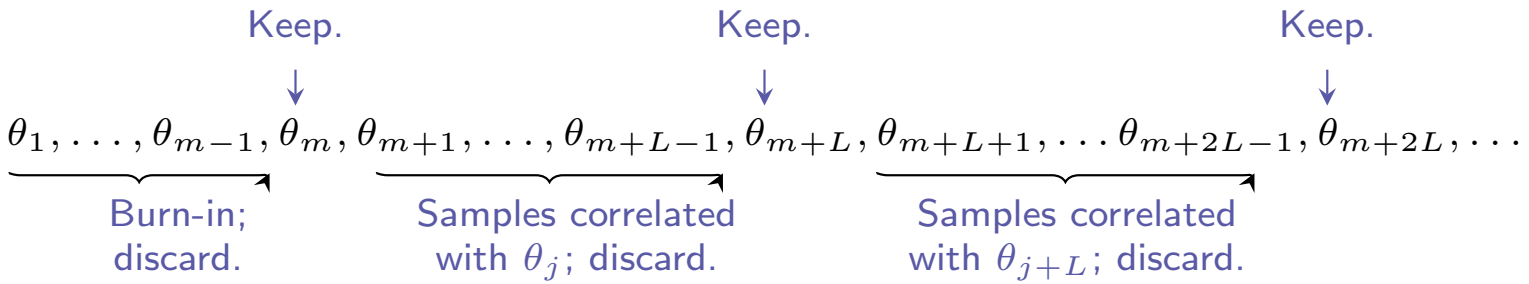
- Hence, we certainly accept if the second term is larger than 1, i.e. if

$$q(\theta_i|\theta_{i+1})p(\theta_{i+1}) > q(\theta_{i+1}|\theta_i)p(\theta_i) .$$

- That means:
  - We always accept the proposal θ_{i+1} if it *increases* the probability under p.
  - If it *decreases* the probability, we still accept with a probability which depends on the difference to the current probability.
- Interpretation as ascent:
  - MH resembles an ascent algorithm on p, which *tends* to move in the direction of increasing probability p.
  - However:
    - The actual steps are chosen at random.
    - The sampler can move "downhill" with a certain probability.
    - When it reaches a local maximum, it does not get stuck there.

Summary: MCMC with MH

- MCMC samplers construct a MC with invariant distribution p.
- The MH kernel is one generic way to construct such a chain from p and a proposal distribution q.
- Formally, q does not depend on p (but arbitrary choice of q usually means bad performance).
- We have to discard an initial number m of samples as burn-in to obtain samples (approximately) distributed according to p.
- After burn-in, we keep only every Lth sample (where $L = \text{lag}$) to make sure the θ_i are (approximately) independent.



MCMC for gp model selection

- Remember, MCMC works well (often slowly) for integrals like:

$$\mathbb{E}(h(\theta)) = \int_{\Theta} h(\theta) \frac{P(D|\theta)P(\theta)}{P(D)} d\theta.$$

- It does not work directly for the seemingly similar integral:

$$P(D) = \int_{\Theta} P(D|\theta)P(\theta) d\theta.$$

- That was why we introduced AIS in the first place.

- Resulting algorithm for calculating $P(D)$ with AIS and MCMC...

AIS with MCMC

Algorithm 1 Model evidence $P(D)$ with AIS and MH

Input: schedule $\alpha_0 = 0 < ... < \alpha_K = 1$
for $r \leftarrow 1, ..., R$ **do**
  $\theta_0 \sim p(\theta)$
  for $k \leftarrow 1, ..., K$ **do**
    $\theta_k \sim p(D|\theta)^{\alpha_k} p(\theta)$ using MH
    $\log \left(\frac{Z_{\alpha_k}}{Z_{\alpha_{k-1}}} \right) \approx (\alpha_k - \alpha_{k-1}) \log p(D|\theta_k)$
  end for
  $Z_r = \sum_k \log \left(\frac{Z_{\alpha_k}}{Z_{\alpha_{k-1}}} \right)$
end for
return $\log Z = \log \left(\frac{1}{R} \sum_r Z_r \right)$.

- Note single MH sample at each α_k.
- Any MCMC should work here (HMC is popular).
- R runs for variance reduction.

Editorial remarks

- MCMC is a heavily used tool in probabilistic machine learning.
- None of this section is GP specific... but very GP relevant.
- Calculating normalizing constants is also a hugely important topic.
- AIS with MCMC is approximately the gold standard.
- Sampling will come up again and again, so it's good to have it in hand.

Outline

Sampling hyperparameters in GP: [MA10]

- We wish to sample from: $P(\theta, f|D) = \frac{1}{Z} P(D|f)P(f|\theta)P(\theta)$.

  intractable integral over latents $\rightarrow$ sample those latents... see AIS

- This is a GP, so $P(f|\theta) = \mathcal{N}(f; 0, K_\theta)$.

- $P(D|f)$ is not assumed to be gaussian (much more in §03).

  ...what do we do if it were gaussian?

- For a fixed f, we can use MCMC out of the box [MA10, Alg. 1].

- That MCMC rejection kernel is:

$$A(\theta'|\theta) := \min\left\{1, \frac{q(\theta|\theta')P(\theta')\mathcal{N}(f; 0, K_{\theta'})}{q(\theta'|\theta)P(\theta)\mathcal{N}(f; 0, K_\theta)}\right\}$$

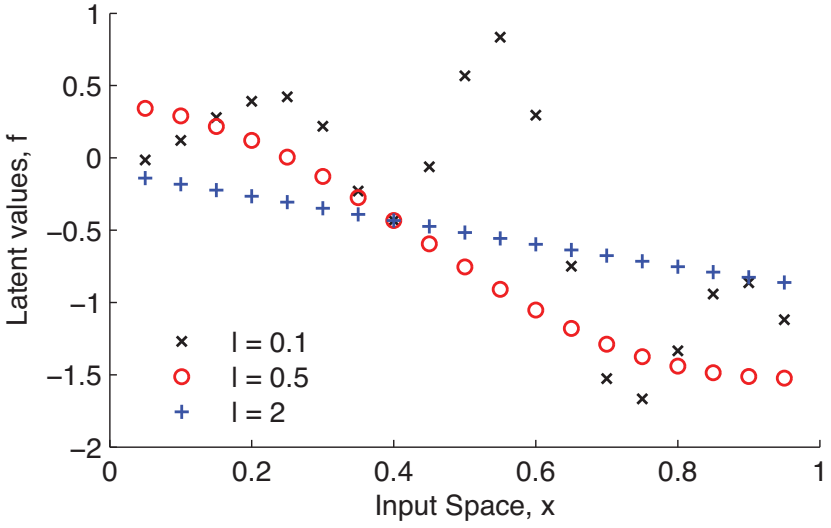
  what again is q here?

  whither $P(D|f)$?

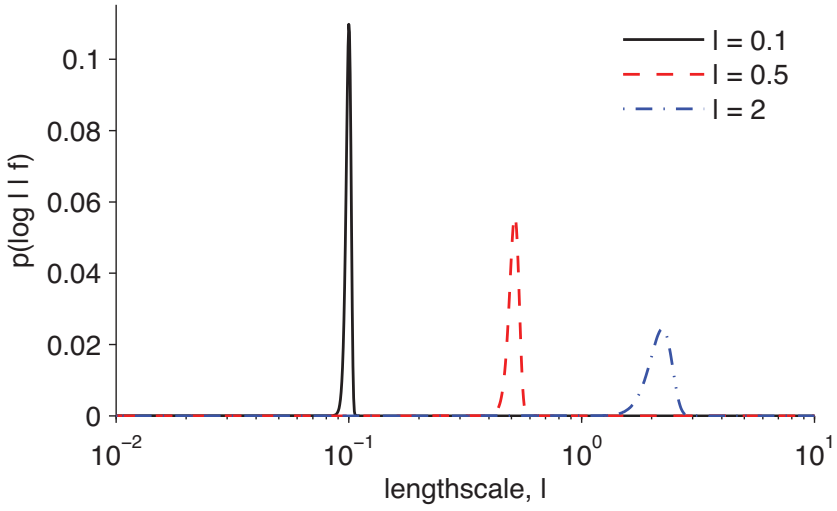
- However...

Sampling hyperparameters in GP: [MA10]

- One is then tempted to alternate:
  - For a fixed f, we can use MCMC out of the box [MA10, Alg. 1].
  - For a fixed θ, sample a new f.
- This alternate sampling scheme will converge very poorly.



(a) Prior draws



(b) Lengthscale given $\mathbf{f}$

- This issue arises from the strong coupling between f and θ.

Sampling hyperparameters in GP: [MA10]

- The previous issue is caused by strong coupling between f and θ.
- The (usual) reparameterization trick with $f = L_\theta \nu$ for $K_\theta = L_\theta L_\theta^\top$:

$$P(f|\theta)P(\theta) = \mathcal{N}(f; 0, K_\theta)P(\theta) = \mathcal{N}(\nu; 0, I)\frac{1}{|L_\theta|}P(\theta).$$

- Now, importantly, for a fixed ν, MCMC on θ draws a new θ' and $f' = L_{\theta'}\nu$.
- That MCMC rejection kernel is [MA10, Alg. 2]

$$A(\theta'|\theta) := \min\left\{1, \frac{q(\theta|\theta')P(\theta')P(D|f' = L_{\theta'}\nu)}{q(\theta'|\theta)P(\theta)P(D|f = L_\theta\nu)}\right\}.$$

- Working through details of [MA10, Eq. 6] will clarify that kernel.

Sampling hyperparameters in GP: [MA10]

- Better, but still not quite enough. Problem:
  - (fundamental reminder – data reduces uncertainty about latents f)
  - [MA10, Alg. 1] (Alg. 2) is optimal in the strong (weak) data limit.
  - Compare (part of) Alg. 1 rejection kernel...

$$\frac{p(\theta'|f,D)}{p(\theta|f,D)} = \frac{P(D|f)\mathcal{N}(f;0,K_{\theta'})P(\theta')}{P(D|f)\mathcal{N}(f;0,K_{\theta})P(\theta)} = \frac{\mathcal{N}(f;0,K_{\theta'})P(\theta')}{\mathcal{N}(f;0,K_{\theta})P(\theta)}$$

  - ...with Alg. 2 rejection kernel:

$$\frac{p(\theta'|\nu,D)}{p(\theta|\nu,D)} = \frac{P(D|f=L_{\theta'}\nu)\mathcal{N}(\nu;0,I)P(\theta')}{P(D|f=L_{\theta}\nu)\mathcal{N}(\nu;0,I)P(\theta)} = \frac{P(D|f=L_{\theta'}\nu)P(\theta')}{P(D|f=L_{\theta}\nu)P(\theta)}.$$

  - f fixed by likelihood $\rightarrow$ Alg. 1 explores posterior (on θ) well.
  - weak data $\rightarrow$ the GP prior is influential, so fixed ν works well.

- Compromise: θ-dependent surrogate observation $g \sim \mathcal{N}(g; f, S_\theta)$.

- Defines approximate posterior on $f \rightarrow$ correctly gauge strength of data.

Sampling hyperparameters in GP: [MA10]

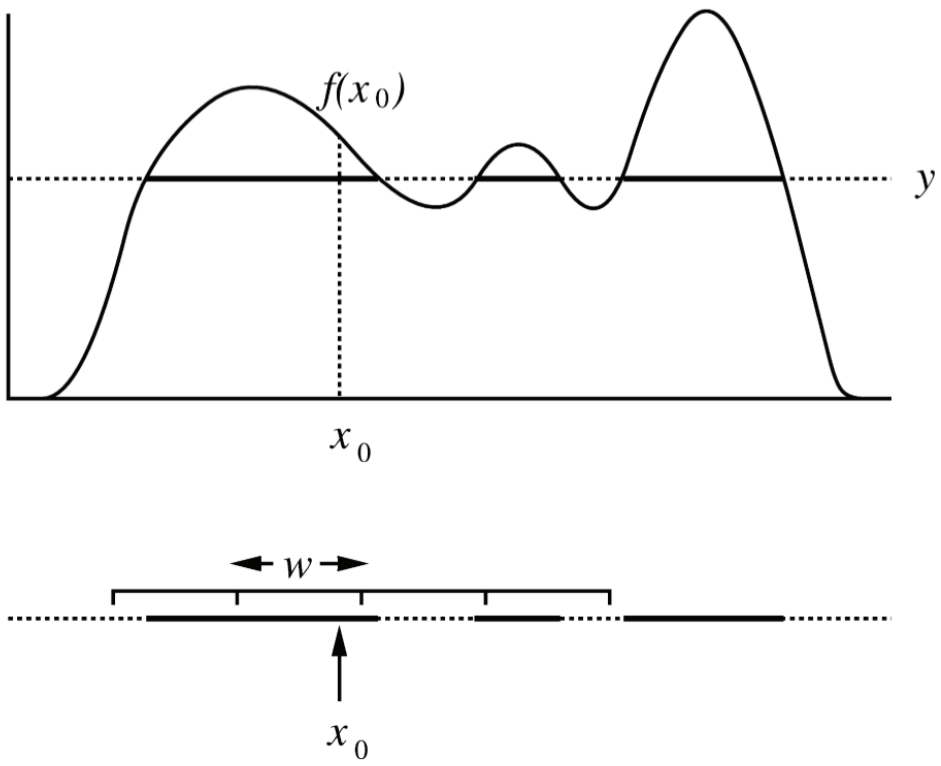
- [MA10, §03] introduces that surrogate $g \sim \mathcal{N}(g; f, S_\theta)$.
- Results in the MCMC rejection kernel is [MA10, Alg. 3]:

$$A(\theta'|\theta) := \min\left\{1, \frac{q(\theta|\theta')P(\theta')P(D|f = L_{\theta'}\nu)\mathcal{N}(g; 0, K_{\theta'} + S_{\theta'})}{q(\theta'|\theta)P(\theta)P(D|f = L_\theta\nu)\mathcal{N}(g; 0, K_\theta + S_\theta)}\right\}.$$

- Notice we being hopeful about the proposal q.
- Solution: use slice sampling [MA10, Alg. 4]...

Slice sampling: an MCMC-like idea

- Recall the idea of MCMC: wander Θ, biasing towards higher $P(\theta|D)$.
- Slice sampling tries to draw uniformly from $(\theta, P(\theta|D))$. Good idea?
  - Uniform draws over that volume are draws $\theta \sim P(\theta|D)$.
  - The desired high-density bias will necessarily exist.



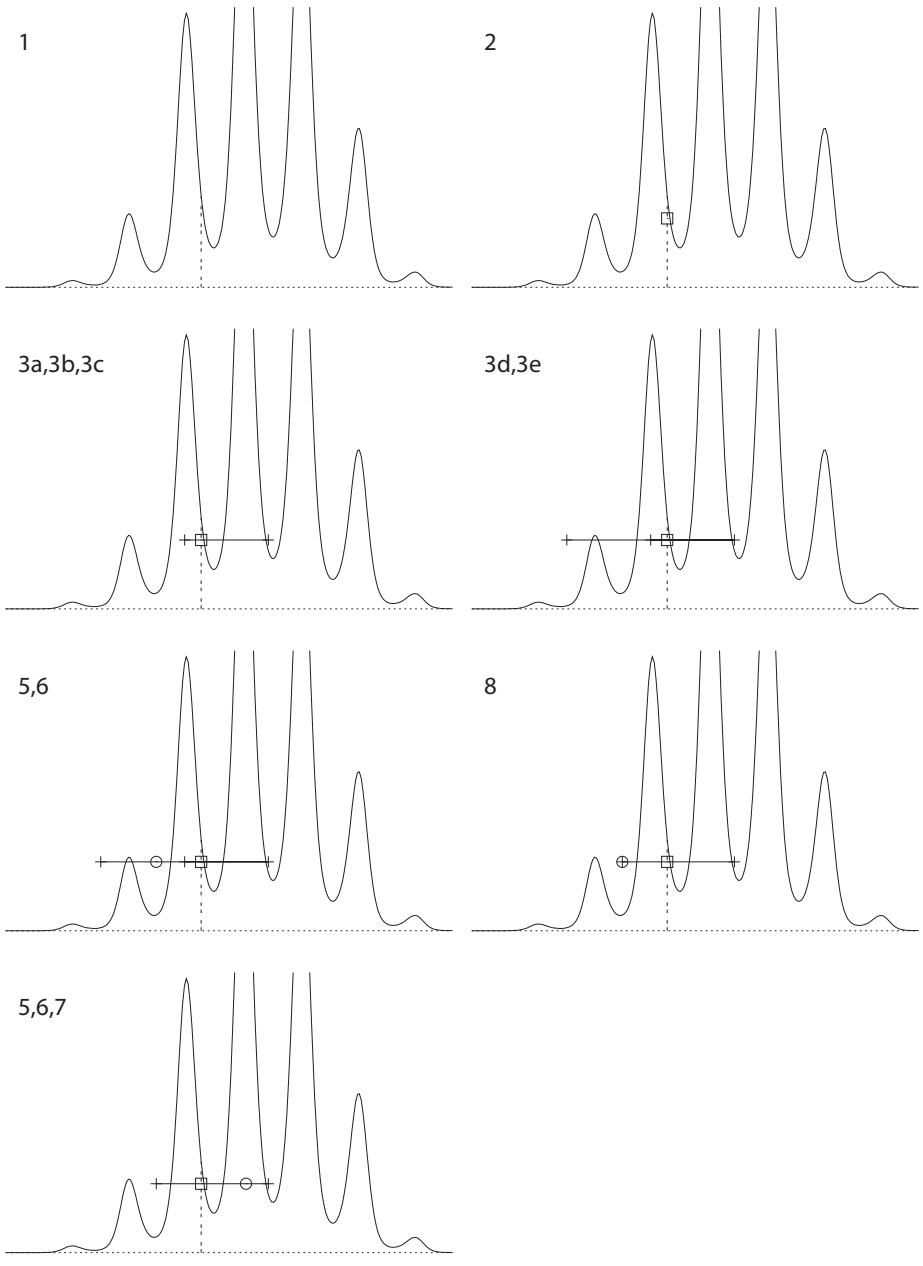
- Original is [Nea03]; I prefer [Mac03, ch. 29]; important for GP: [MAM09].

Slice sampling: an MCMC-like idea

Algorithm 2 Slice sampling $\theta \to \theta'$

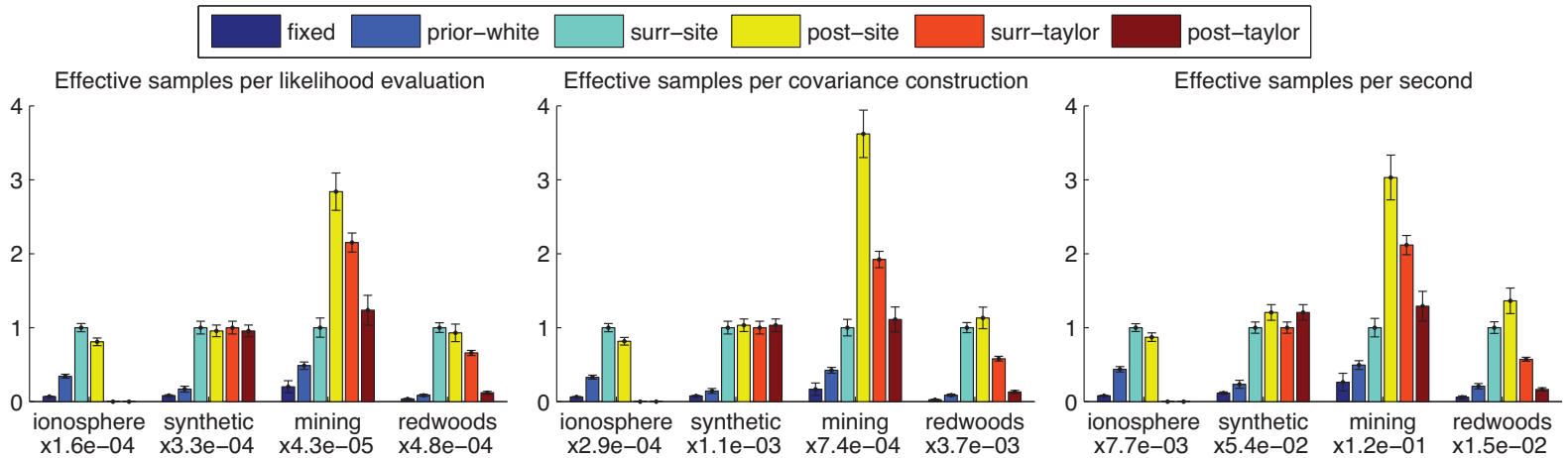
1: Evaluate $P(\theta|D)$
2: Draw vertical $\phi \sim Unif(0, P(\theta|D))$
3: Place slice $[\theta_l, \theta_u]$ around θ
4: **while** **do**
5: $\quad \theta' \sim Unif(\theta_l, \theta_u)$
6: $\quad$ **if** $P(\theta'|D) > \phi$, **then** break
7: $\quad$ **else** change slice interval $[\theta_l, \theta_u]$
8: **end while**
9: **return** sample $\theta' \sim P(\theta|D)$.

- Should feel like rejection sampling.
- No proposal distribution required!
- Still a markov chain operator.
- Some detail hidden in line 7...



Sampling hyperparameters in GP: [MA10]

- Using slice sampling allows us to avoid a proposal distribution q.
- Compare [MA10, Alg. 3] to [MA10, Alg. 4].
- Hopefully less tuning required (in practice, that is indeed often the case).
- Producing the results (and ignoring the taylor parts):



- key:
  - `fixed`: a single fixed latent f... coupling matters.
  - `surr-site`: diagonal S_θ from $P(D_i|f_i)\mathcal{N}(f_i; 0, \{K_\theta\}_{ii})$.
  - `post-site`: diagonal S_θ from $P(\theta|\nu)$.

Takeaways

- Model selection in GP is a research-grade problem.
- Coupling between f and θ causes problems.
- MCMC is essential.
- Reparamerization trick couples MCMC steps.
- Surrogate variance trick gauges likelihood influence.
- Slice sampling makes the implementation less bespoke.

Outline

Approximate integrals over hyperparameters: [GOH14, §3]

- Compare ML-II hyperparameter optimization to our true belief:

$$\begin{aligned} p(f|D) &= \int_\Theta p(f|D,\theta)p(\theta|D)d\theta \\ &\approx p(f|D,\theta_{ML}) \end{aligned}$$

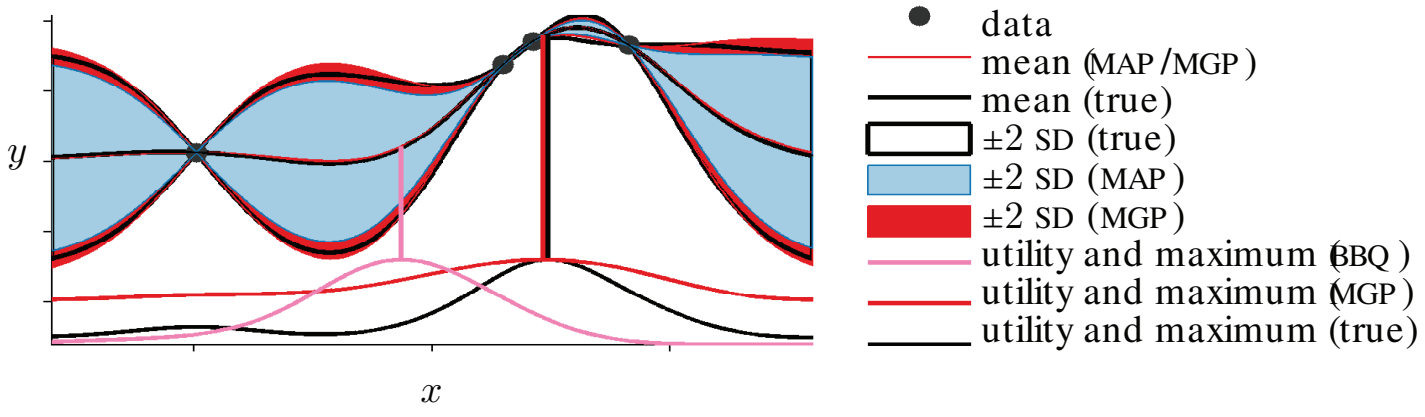
- Posterior uncertainty is **underestimated** by ML-II.
- Even if $p(f|D,\theta)$ is gaussian, $p(f|D)$ is not. Approximation (univariate f!):

$$\begin{aligned} p(f|D) &= \int_\Theta p(f|D,\theta)p(\theta|D)d\theta \\ &\approx \int_\Theta \mathcal{N}(f; a^\top\theta + b, \nu^2)\mathcal{N}(\theta;\hat{\theta},\Sigma)d\theta \end{aligned}$$

- where a, b, ν are chosen to match first and second moments of $p(f|D,\theta)$.
- Leads to a final induced posterior approximation $p(f|D) \approx \mathcal{N}(f;\tilde{m},\tilde{V})$.

Approximate integrals over hyperparameters: [GOH14, §3]

- [GOH14, Eq. 9-19] $\rightarrow$ posterior approx. $p(f|D) \approx \mathcal{N}(f; \tilde{m}, \tilde{V})$.
- Essential takeaway: $p(\theta|D)$ imparts additional randomness onto $p(f|D)$.
- Consequences:



- Note added uncertainty (red) in the posterior
- Absent that, desired inference can be be quite wrong (pink vs. black, red)
- Still an open problem...

References

[GOH14] Roman Garnett, Michael A Osborne, and Philipp Hennig.
Active learning of linear embeddings for gaussian processes.
UAI, 2014.

[HMG15] James Hensman, Alex Matthews, and Zoubin Ghahramani.
Scalable variational gaussian process classification.
AISTATS, 2015.

[KR05] Malte Kuss and Carl Edward Rasmussen.
Assessing approximate inference for binary gaussian process classification.
The Journal of Machine Learning Research, 6:1679–1704, 2005.

[MA10] Iain Murray and Ryan P Adams.
Slice sampling covariance hyperparameters of latent gaussian models.
In *Advances in Neural Information Processing Systems*, pages 1732–1740, 2010.

[Mac03] David JC MacKay.
Information theory, inference and learning algorithms.
Cambridge university press, 2003.

[MAM09] Iain Murray, Ryan Prescott Adams, and David JC MacKay.
Elliptical slice sampling.
arXiv preprint arXiv:1001.0175, 2009.

[Nea03] Radford M Neal.
Slice sampling.
Annals of statistics, pages 705–741, 2003.

[RMC09] Havard Rue, Sara Martino, and Nicolas Chopin.
Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations.
Journal of the royal statistical society: Series b (statistical methodology), 71(2):319–392, 2009.

[RW06] C. E. Rasmussen and C.K.I. Williams.
Gaussian Processes for Machine Learning.
MIT Press, Cambridge, 2006.

[vdVvZ09] Aad W van der Vaart and J Harry van Zanten.
Adaptive bayesian estimation using a gaussian random field with inverse gamma bandwidth.
The Annals of Statistics, pages 2655–2675, 2009.

[VTMW14] Aki Vehtari, Ville Tolvanen, Tommi Mononen, and Ole Winther.
Bayesian leave-one-out cross-validation approximations for gaussian latent variable models.
arXiv preprint arXiv:1412.7461, 2014.