Module 3: Bayesian Nonparametrics

Gaussian Processes for Regression

STAT/BIOSTAT 527, University of Washington

Emily Fox

April 21st-22nd, 2014

Maximum Likelihood Estimation

- Model:

$$y = f(x) + \epsilon \quad \text{where} \quad \epsilon \sim N(0, \sigma^2)$$

assume dist. form on noise

+

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

examine linear basis expansions

- Equivalently,

$$p(y \mid x, \beta, \sigma^2) = N(y \mid f(x), \sigma^2)$$

- For our training data (independent obs) $(x_1, y_1), \ldots, (x_n, y_n)$

$$p(y \mid X, \beta, \sigma^2) = \prod_{i=1}^{n} N(y_i \mid f(x_i), \sigma^2)$$

max log-like (instead of min L_2)

Maximum Likelihood Estimation

$$p(y \mid X, \beta, \sigma^2) = \prod_i N(y_i \mid \beta^T h(x_i), \sigma^2)$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Taking the log

$$\log p(y \mid X, \beta, \sigma^2) = \sum_i -\frac{1}{2}(y_i - \beta^T h(x_i))^2 \underbrace{- \frac{n}{2}\log 2\pi - \frac{n}{2}\log \sigma^2}_{\text{const. wrt } \beta}$$

- Equivalent objective to RSS *(Gaussian log-like loss = L_2 loss)*

- Taking the gradient and setting to zero, we have already shown

$$\hat{\beta}^{ML} = (H^T H)^{-1} H^T y$$

point estimate

$$H = \begin{pmatrix} h_1(x_1) \dots h_M(x_1) \\ \vdots \\ h_1(x_n) \dots h_M(x_n) \end{pmatrix}$$

Bayesian Linear Regression

- More generally, consider a conjugate prior on the basis expansion coefficients:

prior $$p(\beta) = N(\beta \mid \mu_0, \Sigma_0)$$ prior mean + cov.

- Combining this with the Gaussian likelihood function, and using standard Gaussian identities, gives posterior

updated belief about β after obs y

$$p(\beta \mid y) = N(\beta \mid \mu_n, \Sigma_n)$$

posterior (updated) mean + cov. after obs. y

where

$$\mu_n = \Sigma_n(\Sigma_0^{-1}\mu_0 + \sigma^{-2}H^T y)$$

$$\Sigma_n^{-1} = \Sigma_0^{-1} + \sigma^{-2} H^T H$$

incorporates obs. y w/ prior

prior

data dependent

Predictive Distribution

- Predict y^* at new locations x^* by integrating over parameters β

$y_1, \dots, y_n$

$$p(y^* \mid y) = \int p(y^* \mid \beta) p(\beta \mid y) d\beta$$

x^*, x

likelihood

posterior

$$p(\beta \mid y) = N(\beta \mid \mu_n, \Sigma_n)$$

$$p(y \mid x, \beta, \sigma^2) = N(y \mid f(x), \sigma^2)$$

$\beta^T h(x)$

$y^* = h(x^*)^T \beta + \epsilon$

$\beta \mid y \sim N(\mu_n, \Sigma_n)$

$\epsilon \sim N(0, \sigma^2)$

Σ_n fcn of our obs. locations $x_1, \dots, x_n$

$$= N(\mu_n^*(x^*), \Sigma_n^*(x^*))$$

$$\mu_n^*(x^*) = E[y^* \mid y] = \mu_n^T h(x^*)$$

$$\Sigma_n^*(x^*) = \mathrm{cov}(y^* \mid y) = h(x^*)^T \mathrm{cov}(\beta \mid y) h(x^*) + \sigma^2$$

$$= \underbrace{h(x^*)^T \Sigma_n h(x^*)}_{\text{var of } \beta} + \underbrace{\sigma^2}_{\text{var of obs.}}$$

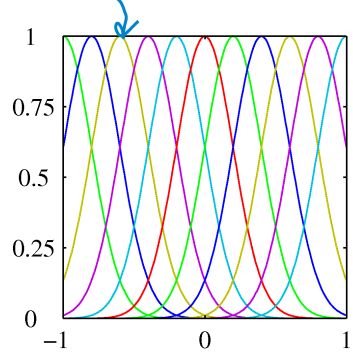
Example: Gaussian Basis Expansion

- Gaussian basis functions:

$$h_j(x) = \exp\left\{ -\frac{(x - \mu_j)^2}{2s^2} \right\}$$

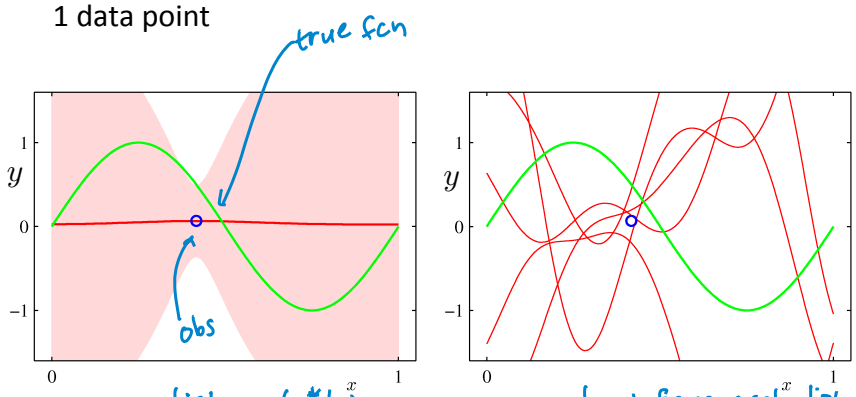
← a choice of a basis exp. with $M=9$

- These are local; a small change in x only affects nearby basis functions. Parameters control location and scale (width)



Example: Gaussian Basis Expansion

- Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point

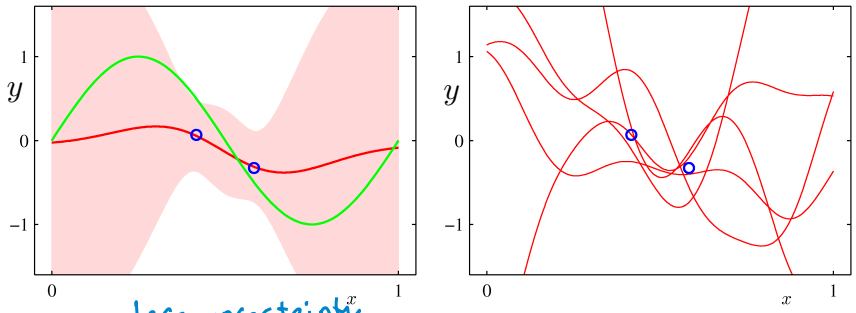


pred. dist. $p(y^*|y)$ shows pointwise uncertainty

draw from pred. dist.

Example: Gaussian Basis Expansion

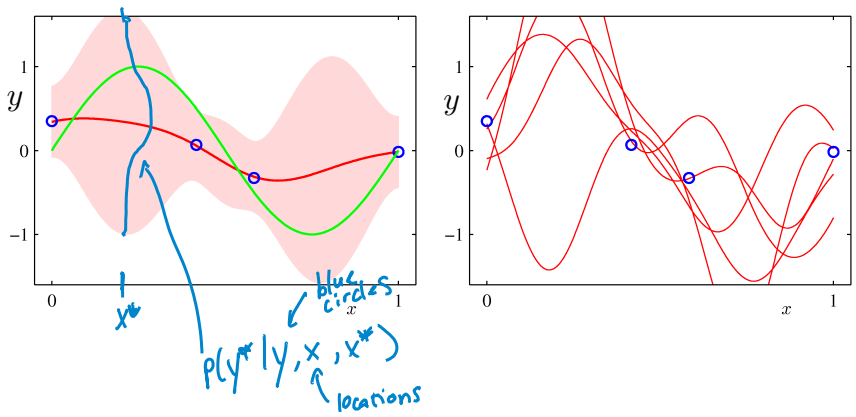
- Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



less uncertainty near obs.

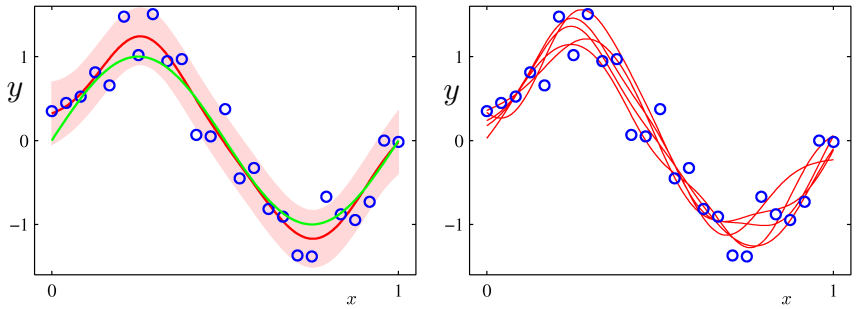
Example: Gaussian Basis Expansion

- Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points

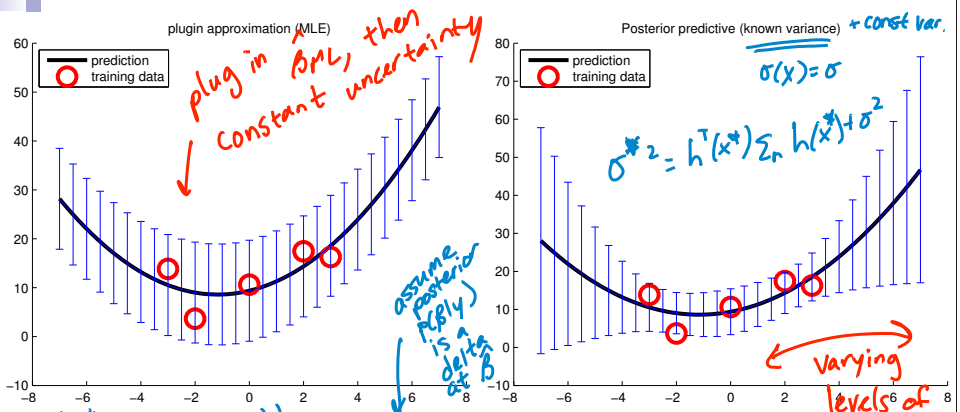


Example: Gaussian Basis Expansion

- Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points



Estimation vs. Predictive Distributions

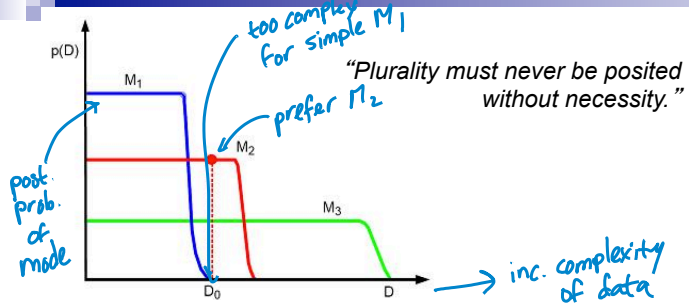


$$p(y^*|x^*, X, y, \sigma^2) \approx \int N(y^*|\beta^T h(x^*), \sigma^2)\, \delta_{\hat{\beta}}(\beta)\, d\beta = N(y^*|\hat{\beta}^T h(x^*), \sigma^2)$$

constant error bars

What M should I use?

Bayesian Ockham's Razor



"Plurality must never be posited without necessity."

William of Ockham

- **Parametric Bayes:** Consider a finite list of possible models, average according to posterior probability (or in practice, just select the most probable)
- **Nonparametric Bayes:** Consider a single infinite model, integrate over parameters when making predictions or infer which finite subset is exhibited in your dataset

@9:00 am

$M \to \infty$

Going Infinite…

model: $f(x) = \phi(x)^T \beta$

Change of notation:
$$h_j(x) \to \phi_j(x)$$

- Nonparametric Gaussian regression: Would like to let the number of basis functions $M \to \infty$ "features"
- *Prior:* $N(\beta \mid 0, \alpha^{-1} I_M)$ ← special case of Gaussian prior
- *Distribution on f:* $f = \Phi\beta$ ← linear comb. of Gaussians ϕ_j ⇒ f is Gauss.

  $(f(x_1), \ldots, f(x_n))^T$ $(\beta_1, \ldots, \beta_M)^T$

  $E[f] = \phi E[\beta] = 0$

  $\mathrm{cov}(f) = \phi \, E[\beta\beta^T] \, \phi^T = \frac{1}{\alpha} \phi\phi^T$

  $p(f) = N(f \mid 0, \alpha^{-1} \phi\phi^T)$

- Gaussian process models replace explicit basis function representation with a direct specification in terms of a ***positive definite kernel function***

Mercer Kernel Functions

- Distributions are of the form
$$p(f) = N(f \mid 0, \alpha^{-1} \underbrace{\Phi\Phi^T}_{n \times n}) = N(f \mid 0, K)$$
$n \times M$ $M \times n$ $n \times n$ matrix

  where the ***Gram matrix*** *K* is defined as
$$K_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$
kernel fn → dim M

- *K* is a ***Mercer kernel*** if the Gram matrix is positive definite for any *n* and any $x_1, \ldots, x_n$

Note: K is nxn matrix regardless of M (dim of basis)

Example of the "kernel trick"

Mercer's Theorem

- If K is positive definite, we can compute the eigendecomp:

$$K = U^T \Lambda U$$

- Then $K_{ij} = (\Lambda^{1/2} U_{\cdot i})^T (\Lambda^{T/2} U_{\cdot j})$
- Define $\phi(x) = \Lambda^{\frac{1}{2}} U_{\cdot i}$ so that

$$K_{ij} = \phi(x_i)^T \phi(x_j)$$ M dim

- If a kernel is Mercer, there exists a function $\phi : \mathcal{X} \to \mathbb{R}^{M}$ s.t.

$$\underline{\underline{\kappa(x,x')}} = \phi(x)^T \phi(x')$$

M might be infinite

Hard to show in general given $\kappa(x,x')$

Example Mercer Kernels

- Example #1: (non-stationary) ***polynomial kernel***

$$\kappa(x,x') = (\gamma x^T x' + r)^{p}$$

- For p=2, γ = r = 1,

$$(1 + x^T x')^2 = (1 + x_1 x_1' + x_2 x_2')^2$$
$$= 1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2'$$

- This can be written as $\phi(x)^T \phi(x')$, with

$$\phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2]$$

  - Equivalent to working in a 6-dimensional feature space $M=6$
  - For general p, basis contains all terms up to degree p
- Example #2: ***Gaussian kernel***

$$\kappa(x,x') = \exp\left(-\frac{1}{2}(x-x')^T \Sigma^{-1} (x-x')\right) \star$$

  - Feature map lives in an infinite-dimensional space

Gaussian Processes

$\cancel{f(x) = \phi(x)^T \beta}$ ← prior on β

work $f(x)$ ← prior on f

- Dispense of parametric view (prior on β) and consider prior on functions themselves (prior on *f*)
- Seems hard, but we have shown that it is feasible when we look at a finite set of values $x_1, \ldots, x_n$

$$p(f) = N(f \mid 0, K)$$

← dist on $\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

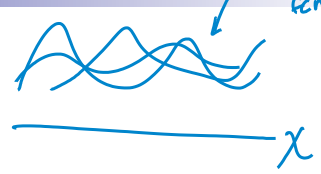
- Defined by a *Mercer kernel*
- More generally, a ***Gaussian process*** provides a distribution over functions

dist on $f(\cdot)$

Gaussian Processes

- Distribution on functions
  - $f \sim GP(m,\kappa)$
    - m: mean function
    - κ: covariance function

GP is a dist on fcns



$\Updownarrow$ iff $\forall n$ and any $x_1, \ldots, x_n$

  - $p(f(x_1), \ldots, f(x_n)) \sim N_n(\mu, K)$
    - $\mu = [m(x_1),\ldots,m(x_n)]$
    - $K_{ij} = \kappa(x_i,x_j)$

- Idea: If x_i, x_j are similar according to the kernel, then $f(x_i)$ is similar to $f(x_j)$

κ: covariance function

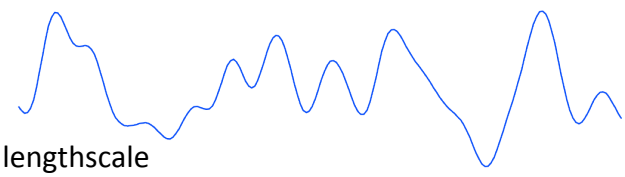
Example: squared exp (SE) or Gauss. kernel

$$\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x - x')^2\right)$$

var

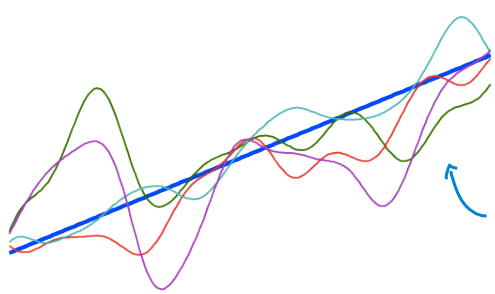
length scale or bandwidth

High lengthscale



Low lengthscale

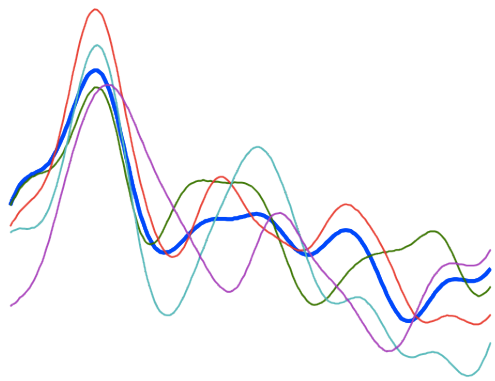
m: mean function



linear trend m(x)

$f \sim GP(m, K)$

m: mean function

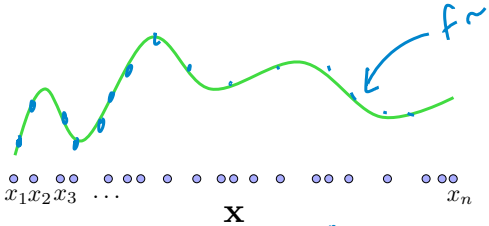


can define more complicated m(x), but typically people choose m(x)=0

Induced Multivariate Gaussian

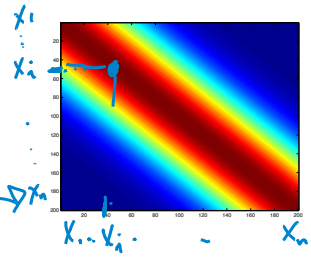
- Evaluating the GP-distributed function at any set of locations, we have

$f \sim GP(0, K)$



corr. drops w/ distance from x_i

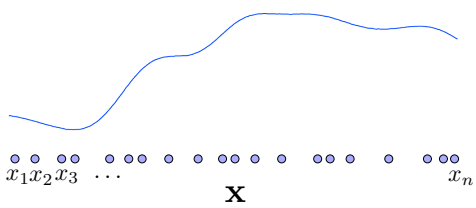
$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim N(0, \underset{n \times n}{K})$$

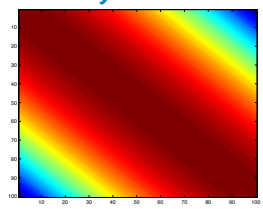


Induced Multivariate Gaussian

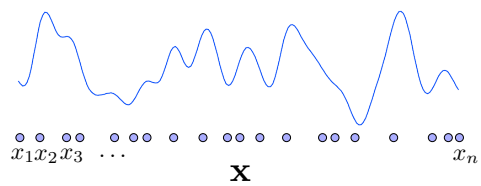
- Comparing length-scales:

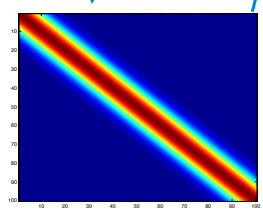
corr. drops slowly





corr. drops rapidly





2D Gaussian Processes

SE example in 2D

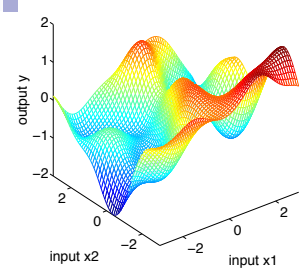
$$\kappa(x_p, x_q') = \sigma_f^2 \exp\left(-\frac{1}{2}(x_p - x_q')^T M (x_p - x_q')\right)$$

2x2

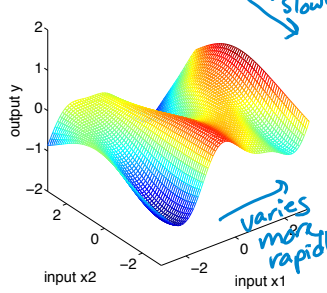
$M = \ell^{-2} I$

$k(x,x') = e^{-\frac{\|x-x'\|^2}{2\ell^2}}$

radial basis fcn (RBF)



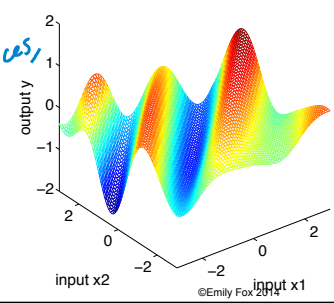
vary slowly



varies more rapidly

other choices, too

Coupling dim.



diff length scales to diff. dims

$M = \begin{bmatrix} \frac{1}{\ell_1^2} & 0 \\ 0 & \frac{1}{\ell_2^2} \end{bmatrix}$

$\ell_1 = 1 \quad \ell_2 = 3$

GPs for Regression

(up to this point, just background on GPs... now have "data")

- Start with noise-free scenario: directly observe the function
- Training data $\mathcal{D} = \{(x_i, f_i), i = 1, \ldots, n\}$
- Test data locations X^* → predict f^*
- Jointly, we have by defn of GP, if $f(\cdot) \sim GP(m, K)$ then

$$\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \quad \begin{pmatrix} f \\ f^* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$$

$(m(x_1), \ldots, m(x_n))$ $\quad K(X, X^*)$

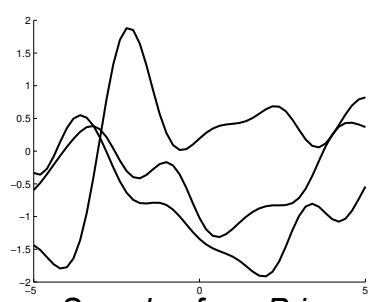
cond. on this $\quad \begin{pmatrix} m(x_1^*) \\ \vdots \\ m(x_{n^*}^*) \end{pmatrix} \quad K(X^*, X^*)$

- Therefore,

$$p(f^* \mid X^*, X, f) = N\left(f^* \mid \mu_* + K_*^T K^{-1}(f - \mu), \; K_{**} - K_*^T K^{-1} K_*\right)$$

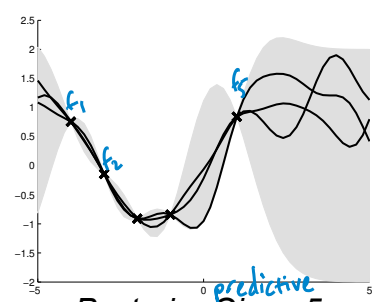
training obs.)

1D Noise-Free Example



Samples from Prior

$$\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x - x')^2\right)$$



Posterior predictive Given 5 Noise-Free Observations

- Interpolator, where uncertainty increases with distance
- Useful as a computationally cheap proxy for a complex simulator
  - Examine effect of simulator params on GP predictions instead of doing expensive runs of the simulator

GPs for Regression

- Noisy scenario: observe a noisy version of underlying function
$$y = f(x) + \epsilon \quad \epsilon \sim N(0, \sigma_y^2)$$
  - Not required to interpolate, just come "close" to observed data
$$\mathrm{cov}(y|X) = \mathrm{cov}(f) + \mathrm{cov}(\epsilon) = K + \sigma_y^2 I_n \triangleq K_y$$
$[y_1, \ldots, y_n]^T$
- Training data $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Test data locations X^* → predict f^*
- Jointly, we have $\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$

for simplicity

as before

cond. on this

$K(X^*, X^*)$ as before

- Therefore, $p(f^* \mid X^*, X, y) = N(f^* \mid K_*^T K_y^{-1} y, K_{**} - K_*^T K_y^{-1} K_*)$

GPs for Regression

$$p(f^* \mid X^*, X, y) = N(K_*^T K_y^{-1} y, K_{**} - K_*^T K_y^{-1} K_*)$$

- For a single point x^*

$$p(f^* \mid \cancel{X^*}, X, y) = N(k_*^T K_y^{-1} y, k_{**} - k_*^T K_y^{-1} k_*)$$

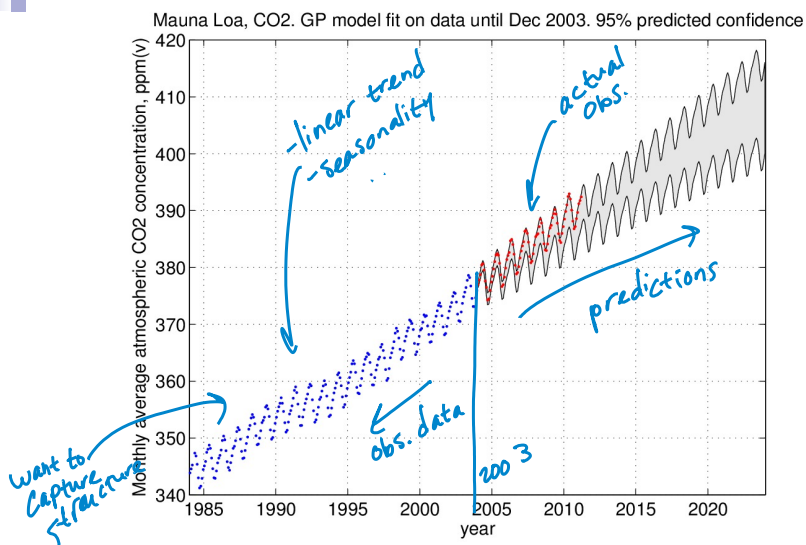
so

$$\bar{f}^* = k_*^T K_y^{-1} y = \sum_{i=1}^{n} \alpha_i K(x_i, x^*)$$

predictive mean

will see this later

CO2 Concentration Over Time



Mauna Loa Observatory in Hawaii, analyzed by Rasmussen & Williams 2006

Mixing Kernels for CO2 GP Analysis

Smooth global trend

$$\kappa_1(x,x') = \theta_1^2 \exp\left(-\frac{(x-x')^2}{2\theta_2^2}\right)$$

Seasonal periodicity

$$\kappa_2(x,x') = \theta_3^2 \exp\left(-\frac{(x-x')^2}{2\theta_4^2} - \frac{2\sin^2(\pi(x-x'))}{\theta_5^2}\right)$$

Medium term irregularities

$$\kappa_3(x,x') = \theta_6^2 \left(1 + \frac{(x-x')^2}{2\theta_8\theta_7^2}\right)^{-\theta_8}$$

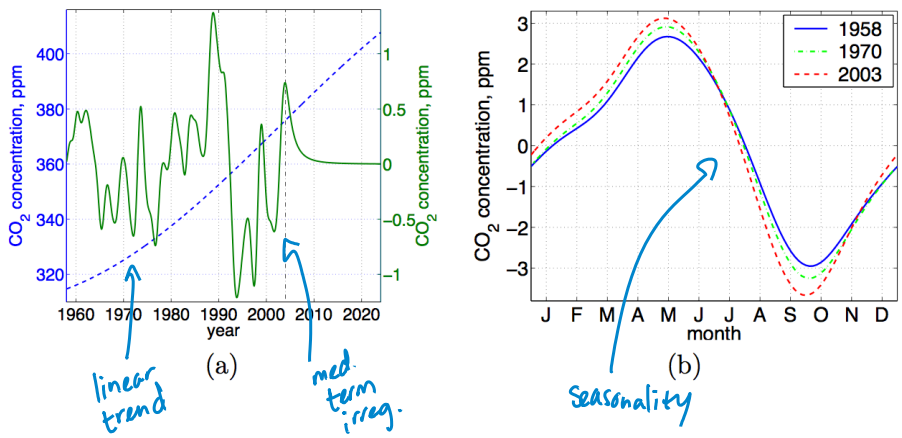
Correlated Observation Noise

$$\kappa_4(x_p,x_q) = \theta_9^2 \exp\left(-\frac{(x_p-x_q)^2}{2\theta_{10}^2}\right) + \theta_{11}^2\delta_{pq}$$

Fun fact:
- sum of kernels is a kernel
- product of kernels is a kernel

$K = K_1 + K_2 + K_3 + K_4$

CO2 Concentration Over Time



hyperparams θ_j are optimized.

Mauna Loa Observatory in Hawaii, analyzed by Rasmussen & Williams 2006

Estimating Hyperparameters

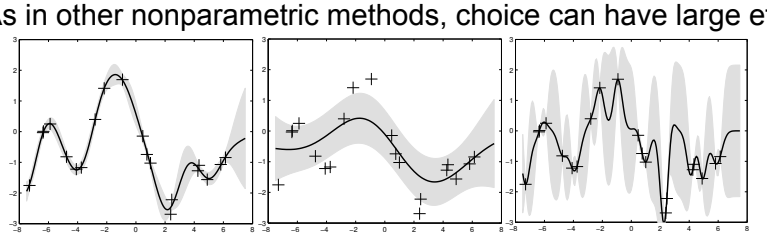
- How should we choose the kernel parameters?
  - Example: squared exponential kernel parameterization

$$\kappa(x,x') = \sigma_f^2 \exp\left(\frac{-1}{2}(x_p - x_q)^T M (x'_p - x'_q)\right) + \sigma_y^2 \delta_{pq}$$

  - Hyperparameters $\theta = \{M, \sigma_f^2, \sigma_y^2\}$
  - As we saw before, can choose

$$M = \ell^{-2} I \quad M = \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_d^{-2}) \quad M = \Lambda\Lambda' + \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_d^{-2}) \ldots$$

- As in other nonparametric methods, choice can have large effect



Estimating Hyperparameters

- Options:
  - #1: Define a grid of possible values and use cross validation

    can be slow...
  - #2: Full Bayesian analysis: Place prior on hyperparameters and integrate over these as well in making predictions

    some challenges in practice
  - #3: Maximize the marginal likelihood

$$p(y \mid X, \theta) = \int p(y \mid f, X) p(f \mid X, \theta) df$$

$$\log p(y \mid X, \theta) =$$

Acknowledgements

Many figures courtesy Kevin Murphy's textbook
Machine Learning: A Probabilistic Perspective,
and Chris Bishop's textbook
Pattern Recognition and Machine Learning

Slides based on parts of the lecture notes of Erik Sudderth for "Applied Bayesian Nonparametrics" at Brown University