[70240413 Statistical Machine Learning, Spring, 2017]

Nonparametric Bayesian Methods (Gaussian Processes)

Jun Zhu

dcszj@mail.tsinghua.edu.cn

http://bigml.cs.tsinghua.edu.cn/~jun

State Key Lab of Intelligent Technology & Systems

Tsinghua University

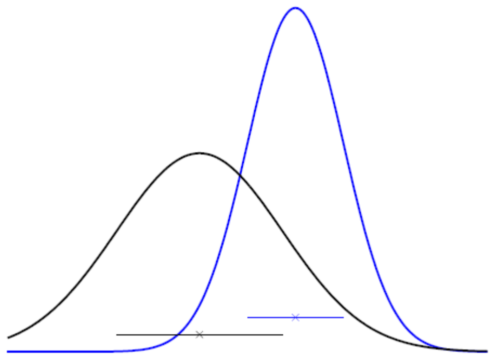
May 23, 2017

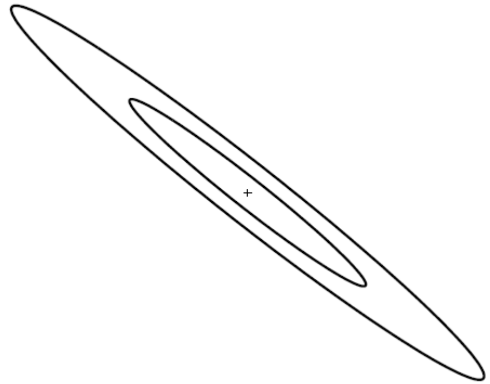
- Today, we talk about Gaussian processes, a nonparametric Bayesian method on the function spaces

- Outline
  - Gaussian process regression
  - Gaussian process classification
  - Hyper-parameters, covariance functions, and more

Recap. of Gaussian Distribution

- Multivariate Gaussian

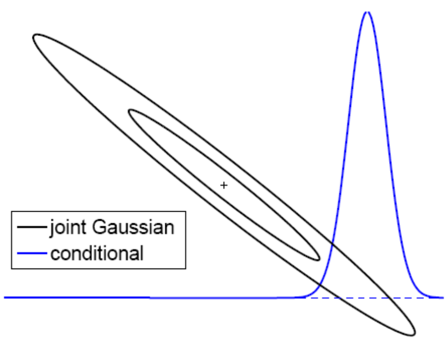


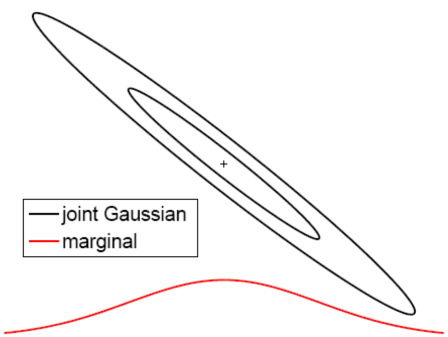


$$p(\mathbf{x}|\mu,\Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2}\exp(-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma^{-1}(\mathbf{x}-\mu))$$

- Marginal & Conditional

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right)$$

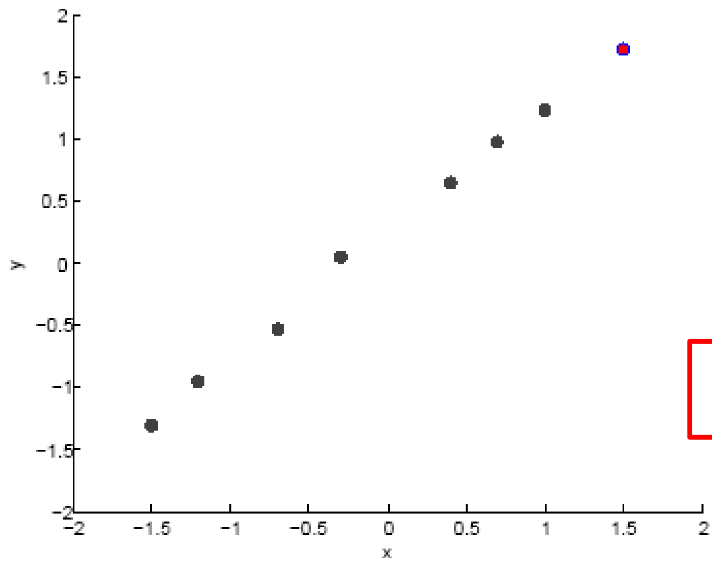




$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mu_x + CB^{-1}(\mathbf{y}-\mu_y), A - CB^{-1}C^\top)$$

$$\mathbf{x} \sim \mathcal{N}(\mu_x, A)$$

A Prediction Task



$$\boxed{f: \mathcal{X} \rightarrow \mathcal{Y}}$$

- Goal: learn a function from noisy observed data
  - Linear $\quad \mathcal{F}_{linear} = \{f: \ f = wx + c, \ \ w, c \in \mathbb{R}\}$
  - Polynomial $\quad \mathcal{F}_{polynomial} = \{f: \ f = \sum_k w_k x^k, \ \ w_k \in \mathbb{R}\}$
  - …

$$\vdots$$

Bayesian Regression Methods

- Noisy observations

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

- Gaussian likelihood function for linear regression $f(x_i) = \mathbf{w}^\top x_i$

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|x_i, \mathbf{w}) = \mathcal{N}(X^\top \mathbf{w}, \sigma_n^2 I)$$

- Gaussian prior (Conjugate)

$$\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$$

- Inference with Bayes' rule
  - Posterior

$$p(\mathbf{w}|X, \mathbf{y}) = \mathcal{N}(\frac{1}{\sigma_n^2} A^{-1} X \mathbf{y}, A^{-1}), \text{ where } A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}$$

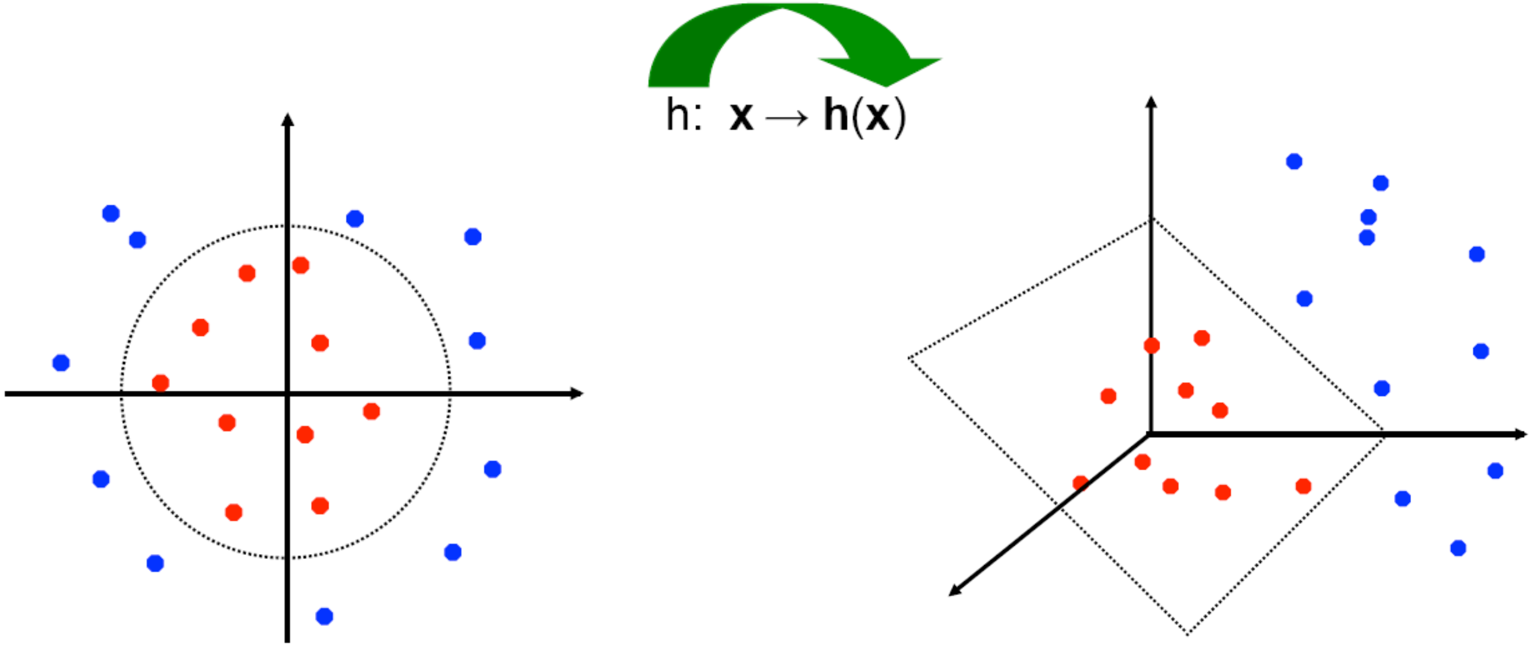
  - Marginal likelihood
  - Prediction

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|X, \mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} = \mathcal{N}(\frac{1}{\sigma_n^2} \mathbf{x}_*^\top A^{-1} X \mathbf{y}, \mathbf{x}_*^\top A^{-1} \mathbf{x}_*)$$

Generalize to Function Space

- The linear regression model can be too restricted.
- How to rescue?
- … by projections (the kernel trick)



Generalize to Function Space

- A mapping function

$$\phi : \mathcal{X} \to \mathbb{R}^N$$

- Doing linear regression in the mapped space

$$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$$

- … everything is similar, with X substituted by $\Phi(X)$

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}\Big(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*)\Big)$$

$$\Phi(X) = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_n)] \quad A = \sigma_n^{-2}\Phi\Phi^\top + \Sigma_p^{-1}$$

Example 1: fixed basis functions

- Given a set of basis functions $\{\phi_h(\mathbf{x})\}_{h=1}^{H}$

$$\phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \cdots \phi_H(\mathbf{x})]^\top$$

  - E.g. 1:

$$\phi_h(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - c_h\|_2^2}{2r^2}\right)$$

  - E.g. 2:

$$\phi_h(\mathbf{x}) = x_i^p x_j^q$$

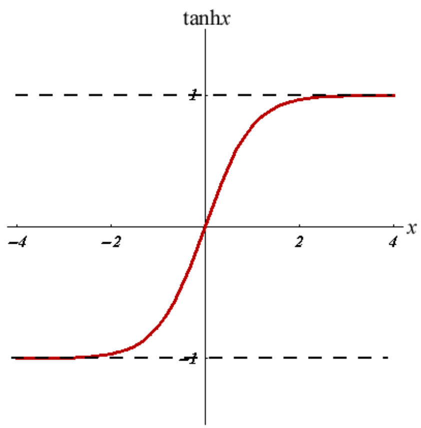
$$\boxed{f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}}$$

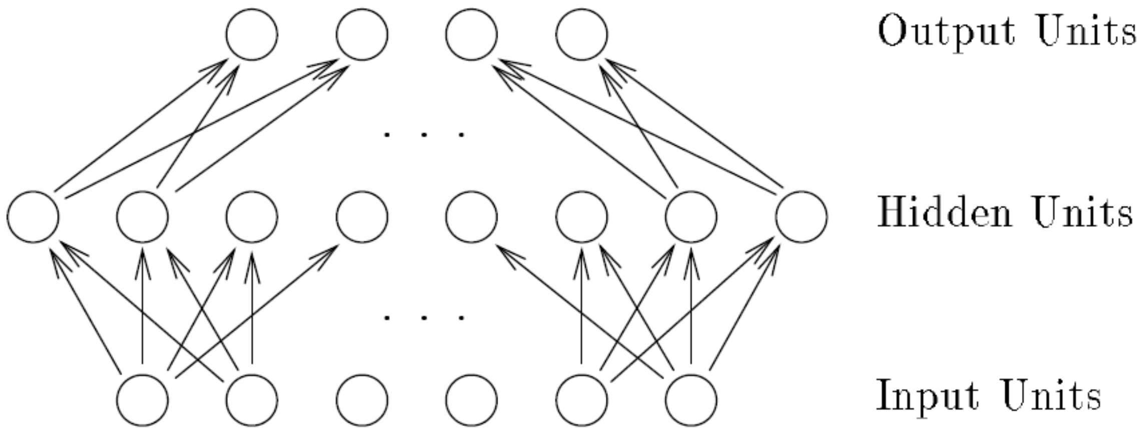
Example 2: adaptive basis functions

- Neural networks to learn a parameterized mapping function
- E.g., a two-layer feedforward neural networks

$$\phi_h(\mathbf{x}) = \tanh\left(\sum_{i=1}^{I} w_{hi}^{(1)} x_i + w_{h0}^{(1)}\right)$$

$$f(\mathbf{x};\mathbf{w}) = \sum_{h=1}^{H} w_h^{(2)} \phi_h(\mathbf{x}) + w_0^{(2)}$$





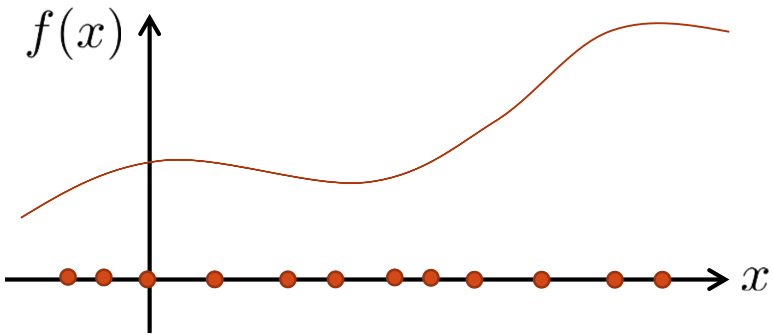
[Figure by Neal]

A Non-parametric Approach

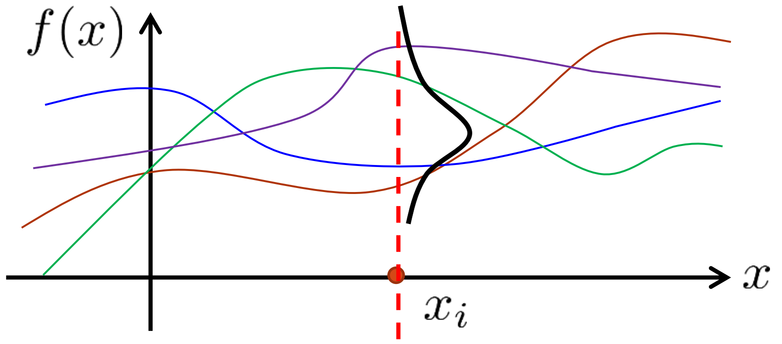
- A non-parametric approach
  - No explicit parameterization of the function
  - Put a prior over all possible functions
  - Higher probabilities are given to functions that are more likely, e.g., of good properties (smoothness, etc.)
  - Manage an uncountably infinite number of functions
  - Gaussian process provides a sophisticated approach with computational tractability

Random Function vs. Random Variable

- A function is represented as an infinite vector with a index set



- For a particular point x_i, $f(x_i)$ is a random variable



Gaussian Process

- A Gaussian process (GP) is a generalization of a multivariate Gaussian distribution to infinitely many variables, thus functions

- **Def**: A stochastic process is Gaussian *iff* for every finite set of indices $x_1, \ldots, x_n$ in the index set $(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n))$ is a vector-valued Gaussian random variable

- A Gaussian distribution is fully specified by the mean vector and covariance matrix

$$\mathrm{f} = (f_1, \cdots, f_n)^\top \sim \mathcal{N}(\mu, \Sigma)$$

- A Gaussian process is fully specified by a mean function and covariance function

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$$

  - Mean function

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

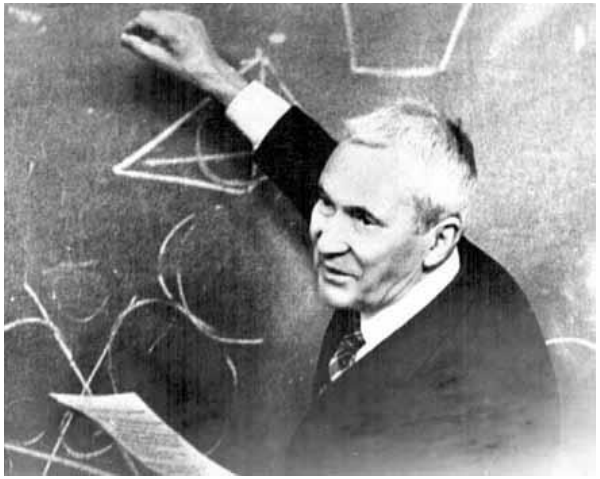
  - Covariance function

$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

Kolmogorov Consistency

- A fundamental theorem guarantees that a suitably "consistent" collection of finite-dim distributions will define a stochastic process
- aka Kolmogorov extension theorem

- Kolmogorov Consistency Conditions
  - Order over permutation
  - Marginalization



Andrey Nikolaevich Kolmogorov
Soviet Russian mathematician
[1903 – 1987]

  - verified with the properties of multivariate Gaussian

Compare to Dirichlet Process

- DP is on random probability measure *P*, i.e., a special type of function
  - Positive, and sum to one!
  - Kolmogorov consistency due to the properties of Dirichlet distribution

- DP: discrete instances (measures) with probability one
  - Natural for mixture models
  - DP mixture is a limit case of finite Dirichlet mixture model

- GP: continuous instances (real-valued functions)
  - Consistency due to the properties of Guassian
  - Good for prediction functions, e.g., regression and classification

Bayesian Linear Regression is a GP

- Bayesian linear regression with mapping functions

$$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w} \qquad \mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$$

- The mean and covariance are

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0$$

$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbb{E}[f(\mathbf{x}) f(\mathbf{x}')] = \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \phi(\mathbf{x}') = \phi(\mathbf{x}) \Sigma_p \phi(\mathbf{x}')$$

- Therefore,

$$f(\mathbf{x}) \sim \mathcal{GP}(0, \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}'))$$

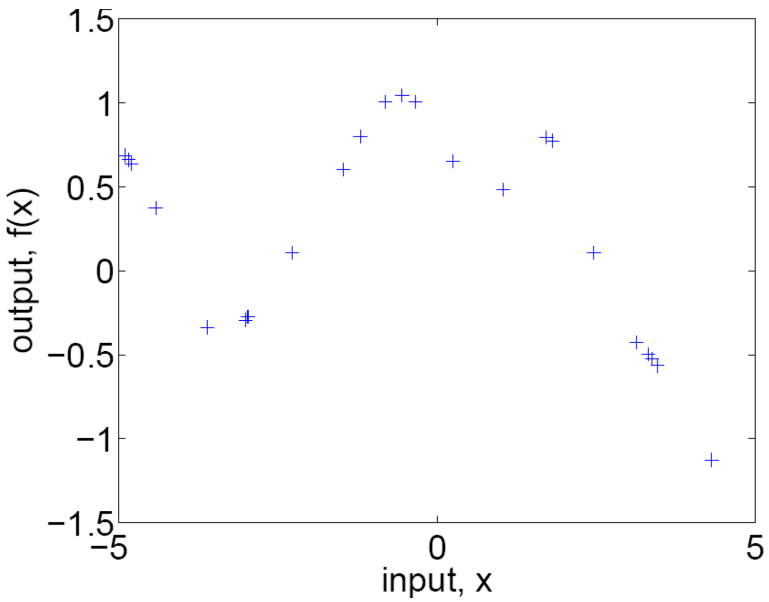
Draw Random Functions from a GP

- Example:

$$p(f(x)) \sim \mathcal{GP}\Big(m(x) = 0, \kappa(x, x') = \exp(-\frac{1}{2}(x - x')^2)\Big)$$

  - For a finite subset

$$(f(x_1), \cdots, f(x_n)) \sim \mathcal{N}(0, \Sigma), \text{ where } \Sigma_{ij} = \kappa(x_i, x_j)$$



Draw Samples from Multivariate Gaussian

- **Task**: draw a set of samples from

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma)$$

- Directly draw is apparently impossible
- A procedure is as follows
  - Cholesky decomposition (aka "matrix square root")

$$\Sigma = LL^\top$$

$$L \text{ is a lower triangular matrix.}$$

  - Generate $\mathbf{y} \sim \mathcal{N}(0, I)$
  - Compute $\mathbf{x} = \mu + L\mathbf{y}$

$$\Rightarrow \quad \mathbb{E}[\mathbf{x}] = \mu, \ \ \mathrm{cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^\top] = L\mathbb{E}[\mathbf{y}\mathbf{y}^\top]L^\top = \Sigma$$

Prediction with Noise-free Observations

- For noise-free observations, we know the true function value

$$\{(\mathbf{x}_i, f_i)\}_{i=1}^{n}$$

- The joint distribution of training output $\mathbf{f}$ and test outputs $\mathbf{f}_*$

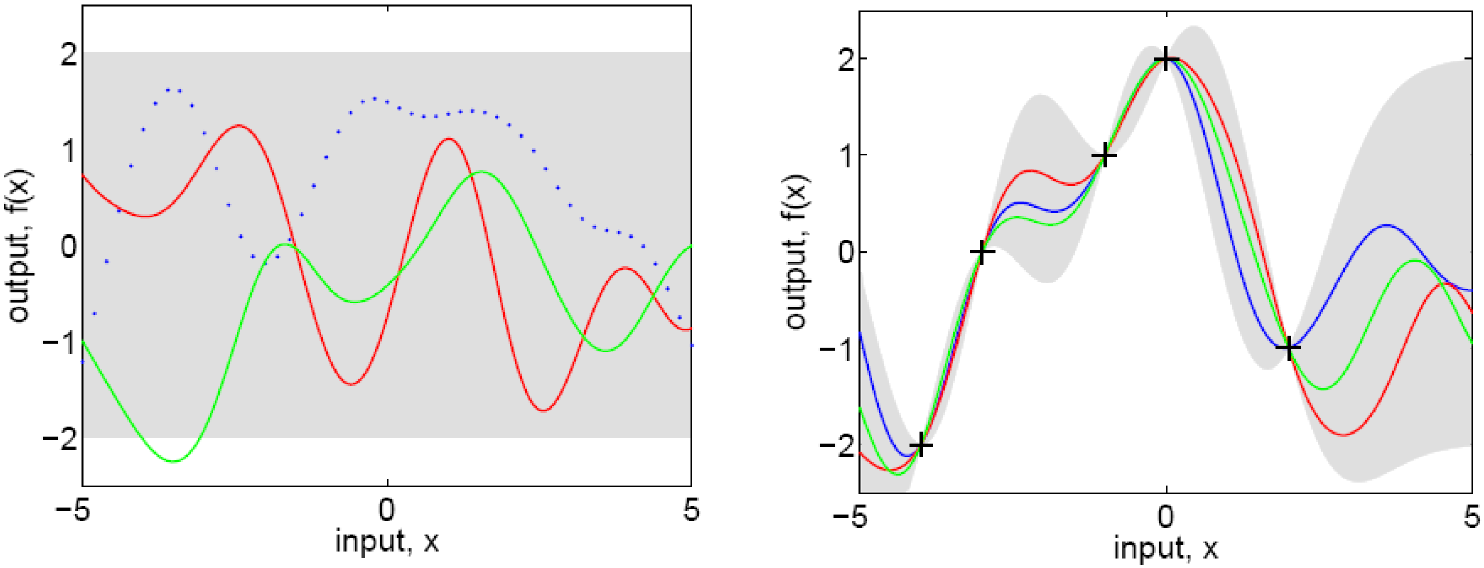
$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

$$\begin{aligned} \mathbf{f}_*|X_*, X, \mathbf{f} \sim \mathcal{N}\Big(& K(X_*,X)K(X,X)^{-1}\mathbf{f}, \\ & K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\Big) \end{aligned}$$

$$\boxed{\begin{gathered} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right) \\ \mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mu_x + CB^{-1}(\mathbf{y} - \mu_y), A - CB^{-1}C^\top) \end{gathered}}$$

Posterior GP

- Samples from the prior and the posterior after observing "+"



(a), prior (b), posterior

  - shaded region denotes twice the standard deviation at each input
- Why the variance at the training points is zero?

Prediction with Noisy Observations

- For noisy observations, we don't know true function values

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n \qquad y_i = f(\mathbf{x}_i) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

$$\Rightarrow \quad \mathrm{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \quad \text{or} \quad \mathrm{cov}(\mathbf{y}) = K(X, X) + \delta_n^2 I$$

- The joint distribution of training output $\mathbf{y}$ and test outputs $\mathbf{f}_*$

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \delta_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

$$\mathbf{f}_* | X_*, X, \mathbf{y} \sim \mathcal{N}\Big(K(X_*, X)[K(X,X) + \delta_n^2 I]^{-1}\mathbf{y}, \\ K(X_*, X_*) - K(X_*, X)[K(X,X) + \delta_n^2 I]^{-1} K(X, X_*)\Big)$$

- Is the variance at the training points zero?

Residual Modeling with GP

- Explicit Basis Function:

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^\top \beta, \text{ where } f(\mathbf{x}) \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}'))$$

  - residual modeling with GP
  - an example of semi-parametric model
  - if we assume a normal prior

$$\beta \sim \mathcal{N}(\mathbf{b}, B)$$

  - we have

$$g(\mathbf{x}) \sim \mathcal{GP}\Big(\mathbf{h}(\mathbf{x})^\top \mathbf{b}, \kappa(\mathbf{x}, \mathbf{x}') + \mathbf{h}(\mathbf{x})^\top B \mathbf{h}(\mathbf{x}')\Big)$$

  - Similarly, we can derive the predictive mean and covariance

Outline

- Introduction
- Gaussian Process Regression
- Gaussian Process Classification

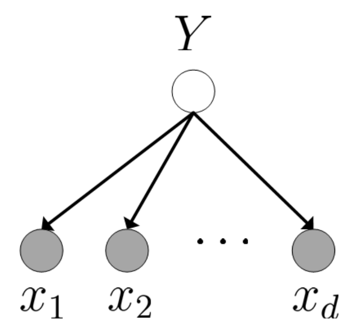
Recap. of Probabilistic Classifiers

- Naïve Bayes (generative models)
  - The prior over classes $p(y)$
  - The likelihood with strict conditional independence assumption on inputs

$$p(x_1, \ldots, x_d | y) = \prod_{i=1}^{d} p(x_i | y)$$

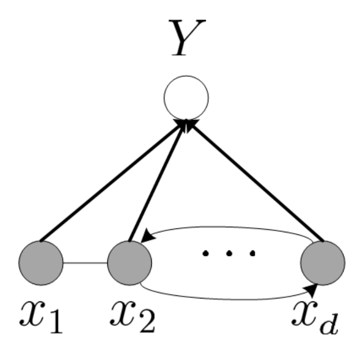
  - Bayes' rule is used for posterior inference

$$p(y|\mathbf{x}) \propto p(y) p(x_1, \ldots, x_d | y)$$



- Logistic regression (conditional/discriminative models)
  - Allow arbitrary structures in inputs

$$p(y|\mathbf{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, y)\}}{\sum_{y'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, y')\}}$$

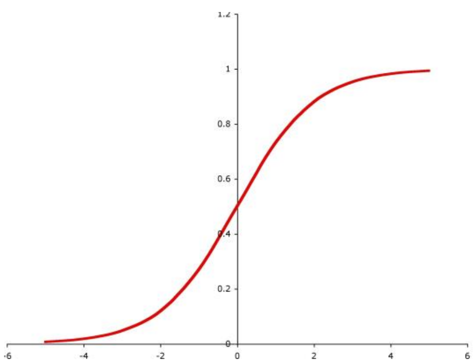


Recap. of Probabilistic Classifiers

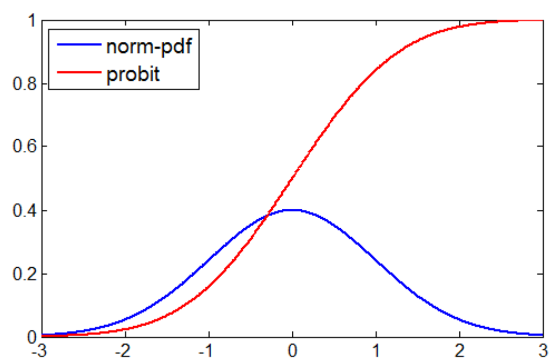
- More on the discriminative methods (binary classification)

$$p(y = +1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x})$$

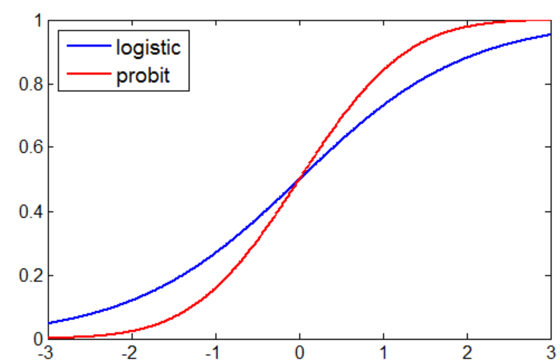
  - σ is the *response function* (the inverse is a *link function*)



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



$$\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x|0, 1)dx$$



comparison

Recap. of Probabilistic Classifiers

- MLE estimation

$$\max_{\mathbf{w}} \quad \log p(\mathbf{y}|X, \mathbf{w})$$

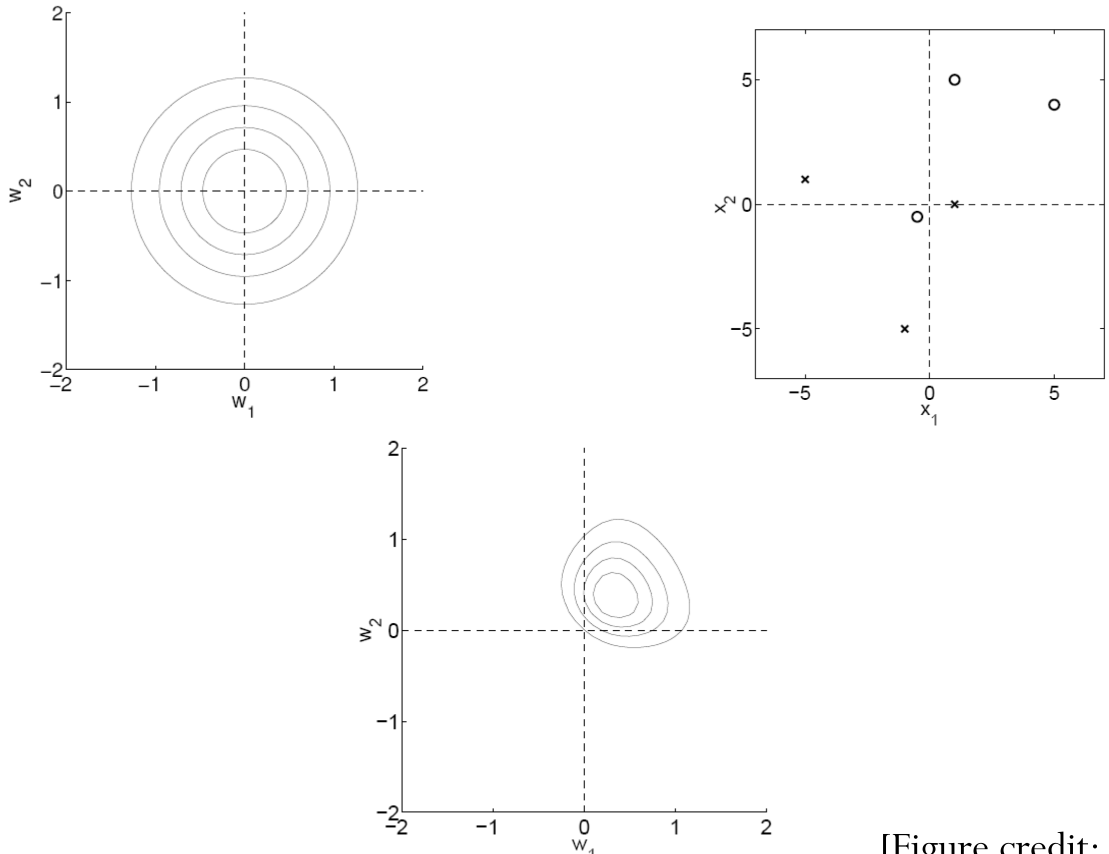
- The objective function is smooth and concave, with unique maximum
- We can solve it using Newton's methods, or conjugate gradient descent

- w goes to infinity for separable case

Bayesian Logistic Regression

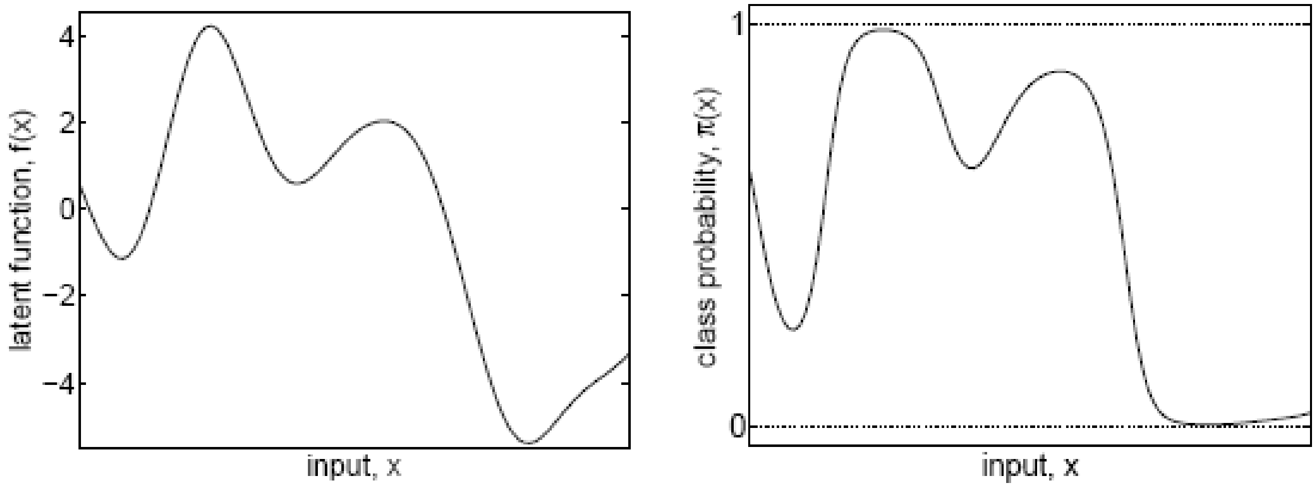
- Place a prior over w

$$p(\mathbf{w}) = \mathcal{N}(0, \Sigma_p)$$



[Figure credit: Rasmussen & Williams, 2006]

Gaussian Process Classification

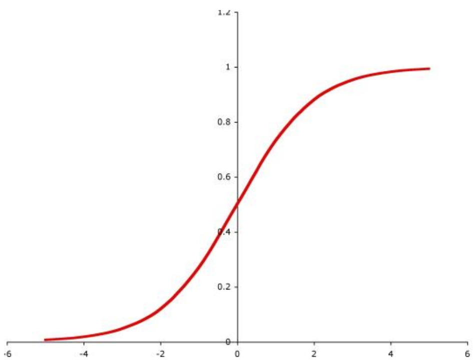


(a) (b)

- Latent function f(x)

$$f(x) \sim GP(\mathrm{m}(x), K(x,x'))$$

$$\pi(x) \triangleq p(y=+1|x) = \sigma(f(x))$$



- Observations are independent given the latent function

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i)$$

Posterior Inference for Classification

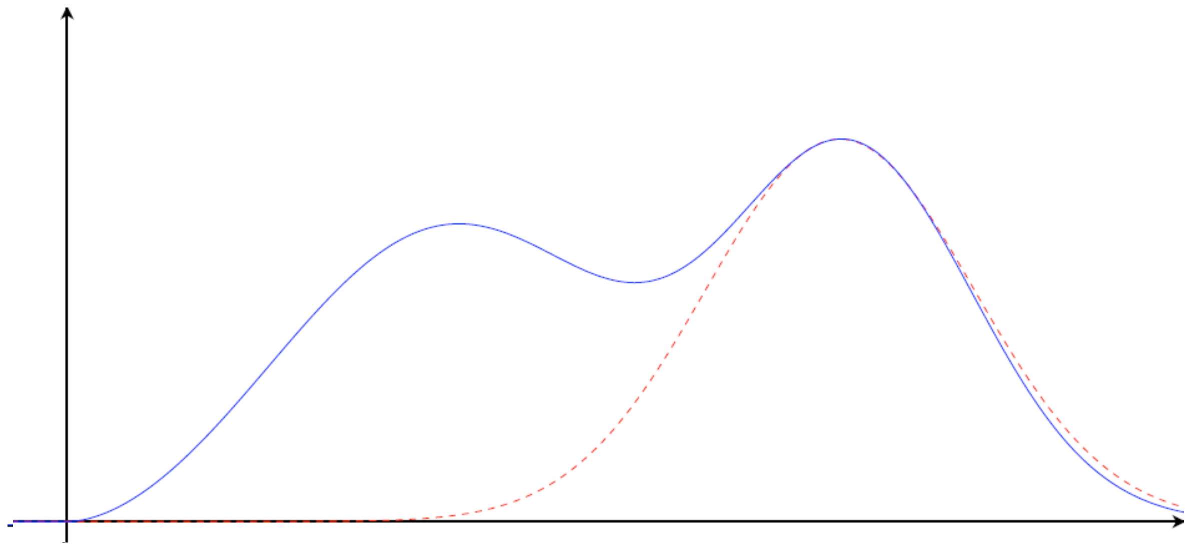
- Posterior (Non-Gaussian)
- Latent value $p(\mathbf{f}|X,\mathbf{y}) = \dfrac{\mathcal{N}(m(\mathbf{x}), K(X,X))}{p(X,\mathbf{y})} \prod_{i=1}^{n} p(y_i|f_i)$
- Predictive distribution

$$p(f_*|X,\mathbf{y},\mathbf{x}_*) = \int p(f_*|X,\mathbf{x}_*,\mathbf{f})p(\mathbf{f}|X,\mathbf{y})d\mathbf{f}$$

$$p(y_* = +1|X,\mathbf{y},\mathbf{x}_*) = \int \sigma(f_*)p(f_*|X,\mathbf{y},\mathbf{x}_*)df_*$$

Laplace Approximation Methods

- Approximating a hard distribution with a “nicer” one



- Laplace approximation is a method using a Gaussian distribution as the approximation
- What Gaussian distribution?

Laplace Approximation Methods

- Approximate the integrals of the form

$$\int_a^b e^{Mf(x)}dx$$

  - assume $f(x)$ has global maximum at x_0
  - then $f(x_0) \geq f(x)$ for any $x \neq x_0$
  - since $e^{Mf(x)}$ growing exponentially with *M*, it's enough to focus on $f(x)$ at x_0

- As *M* increases, integral is well-approximated by a Gaussian

$$\int_a^b e^{Mf(x)}dx \approx \sqrt{\frac{2\pi}{M|\nabla^2 f(x_0)|}} e^{Mf(x_0)} \text{ as } M \to \infty$$

where $\nabla^2 f(x)$ denotes $\nabla\nabla f(x)$

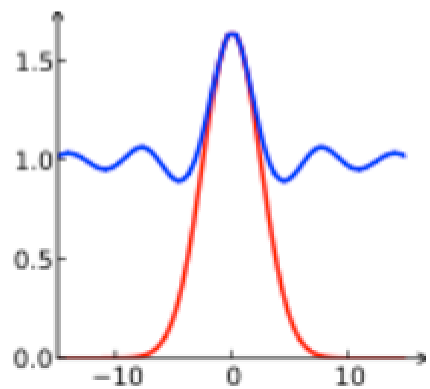
Laplace Approximation Methods

- An example:

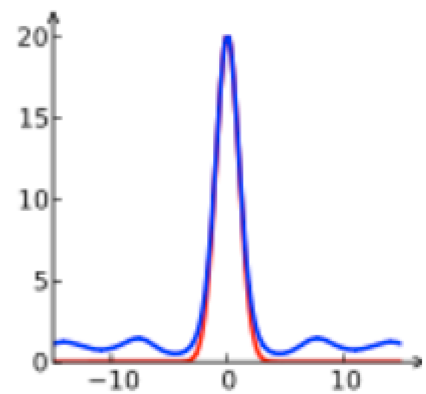
$$f(x) = \frac{\sin x}{x}$$

  - a global maximum is $x_0 = 0$

$e^{Mf(x)}$



$M = 0.5$



$M = 3$

Laplace Approximation Methods

- Deviations by Taylor series expansion

$$f(x) = f(x_0) + \nabla f(x)|_{x=x_0}(x - x_0) + \frac{1}{2}\nabla^2 f(x)|_{x=x_0}(x - x_0)^2 + h.o.t \ldots$$

  - assume that the high-order terms are negligible
  - since $f(x_0)$ is a local maxima, $\nabla f(x)|_{x=x_0} = 0$

- Then, take the first three terms of the Taylor series at x_0

$$f(x) \approx f(x_0) + \frac{1}{2}\nabla^2 f(x)|_{x=x_0}(x - x_0)^2$$

$$\int_a^b e^{Mf(x)}dx = e^{Mf(x_0)} \int_a^b \exp\left(\frac{1}{2}M\nabla^2 f(x)|_{x=x_0}(x - x_0)^2\right)dx$$

$$\text{Let } \sigma^2 = -\frac{1}{M\nabla^2 f(x)|_{x=x_0}}$$

$$\int_a^b e^{Mf(x)}dx = e^{Mf(x_0)} \int_a^b \exp\left(-\frac{1}{2\sigma^2}(x - x_0)^2\right)dx = e^{Mf(x_0}\sqrt{2\pi\sigma^2}$$

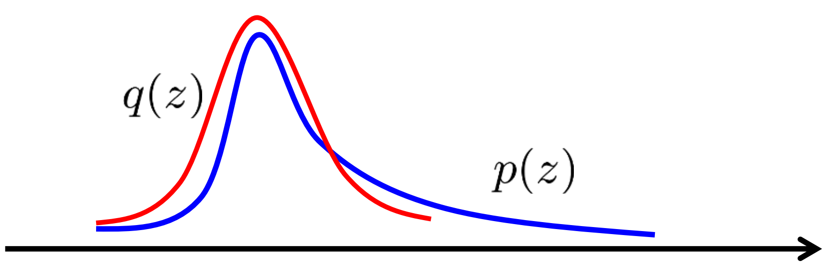
Application: approximate a hard dist.

- Consider single variable z with distribution

$$p(z) = \frac{1}{Z} f(z)$$

  - where the normalization constant is unknown
  - $f(z)$ could be a scaled version of $p(z)$

- Laplace approximation can be applied to find a Gaussian approximation centered on the mode of $p(z)$



Application: approximate a hard dist.

- Doing Taylor expansion in the logarithm space

$$p(z) = \frac{1}{Z} f(z) = \frac{1}{Z} e^{\ln f(z)}$$

  - z_0 is the mode. We have

$$\nabla p(z)|_{z_0} = 0 \qquad \nabla f(z)|_{z_0} = 0 \qquad \nabla \ln f(z)|_{z_0} = 0$$

  - Then, the Taylor series on z_0 is

$$\ln f(z) = \ln f(z_0) - \frac{1}{2} A (z - z_0)^2 \qquad \text{where } A = -\nabla^2 \ln f(z)|_{z=z_0}$$

  - Taking exponential, we have $f(z) \approx f(z_0) \exp\left(-\frac{1}{2} A (z - z_0)^2 \right)$

$$Z \triangleq \int f(z) dz \approx \int f(z_0) \exp\left(-\frac{1}{2} A (z - z_0)^2 \right) dz$$

$$= f(z_0) \sqrt{\frac{2\pi}{A}}$$

$$q(z) = \frac{\tilde{f}(z)}{\tilde{Z}} = \mathcal{N}(z_0, A^{-1})$$

Application: generalize to multivariate

- Task: approximate $p(\mathbf{z}) = \frac{1}{Z} f(\mathbf{z})$ defined over M-dim space
- Find a stationary point $\mathbf{z}_0$, where $\nabla f(\mathbf{z})|_{\mathbf{z}_0} = 0$
- Do Taylor series expansion in log-space at $\mathbf{z}_0$

$$\ln f(\mathbf{z}) = \ln f(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^\top A(\mathbf{z} - \mathbf{z}_0)$$

  - where A is the $M \times M$ Hessian matrix

$$A = -\nabla^2 f(\mathbf{z})|_{\mathbf{z}_0}$$

- Take exponential and normalize

$$f(\mathbf{z}) = f(\mathbf{z}_0) \exp\Big(-\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^\top A(\mathbf{z} - \mathbf{z}_0)\Big)$$

$$q(\mathbf{z}) = \mathcal{N}\Big(\mathbf{z}_0, A^{-1}\Big)$$

Steps in Applying Laplace Approximation

- Find the mode
  - Run a numerical optimization algorithm
  - Multimodal distributions lead to different Laplace approximations depending on the mode considered

- Evaluate the Hessian matrix A at that mode

Approximate Gaussian Process

- Using a Gaussian to approximate the posterior

$$p(\mathbf{f}|X,\mathbf{y}) \approx q(\mathbf{f}|X,\mathbf{y}) = \mathcal{N}(\mathbf{m}, A^{-1})$$

- Then, the latent function distribution

$$q(f_*|X,\mathbf{y},\mathbf{x}_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2), \text{ where}$$

$$\mu_* = \mathrm{k}_*^\top K^{-1}\mathbf{m},\ \sigma_*^2 = \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathrm{k}_*^\top (K^{-1} - K^{-1}A^{-1}K^{-1})\mathrm{k}_*$$

- Laplace method to a nice Gaussian

$$q(\mathbf{f}|X,\mathbf{y}) = \mathcal{N}\left(\mathbf{f}|\hat{\mathbf{f}}, A^{-1}\right) \propto \exp\left(-\frac{1}{2}(\mathbf{f}-\hat{\mathbf{f}})^\top A(\mathbf{f}-\hat{\mathbf{f}})\right)$$

$$\text{where } \hat{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|X,\mathbf{y}) \text{ and } A = -\nabla\nabla \log p(\mathbf{f}|X,\mathbf{y})|_{\mathbf{f}=\hat{\mathbf{f}}}$$

Laplace Approximation for GP

- Computing the mode and Hessian matrix
- The true posterior

$$p(\mathbf{f}|X, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)}{p(\mathbf{y}|X)}$$

  - normalization constant
- Find the MAP estimate

$$\begin{aligned}\psi(\mathbf{f}) &= \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}|X) \\ &= \log p(\mathbf{y}|\mathbf{f}) - \frac{1}{2}\mathbf{f}^\top K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi\end{aligned}$$

  - Take the derivative

$$\nabla\psi(\mathbf{f}) = \nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f}$$

$$\nabla^2\psi(\mathbf{f}) = \nabla^2 \log p(\mathbf{y}|\mathbf{f}) - K^{-1} = -W - K^{-1}$$

Laplace Approximation for GP

- The derivatives of the log posterior are

$$\nabla \psi(\mathbf{f}) = \nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f}$$

$$\nabla^2 \psi(\mathbf{f}) = \nabla^2 \log p(\mathbf{y}|\mathbf{f}) - K^{-1} = -W - K^{-1}$$

  - W is diagonal since data points are independent

- Finding the mode
  - *Existence of maximum*
  - For logistic, we have

How about probit regression? (homework)

$$\nabla_{f_i} \log p(y_i|f_i) = t_i - \pi_i$$

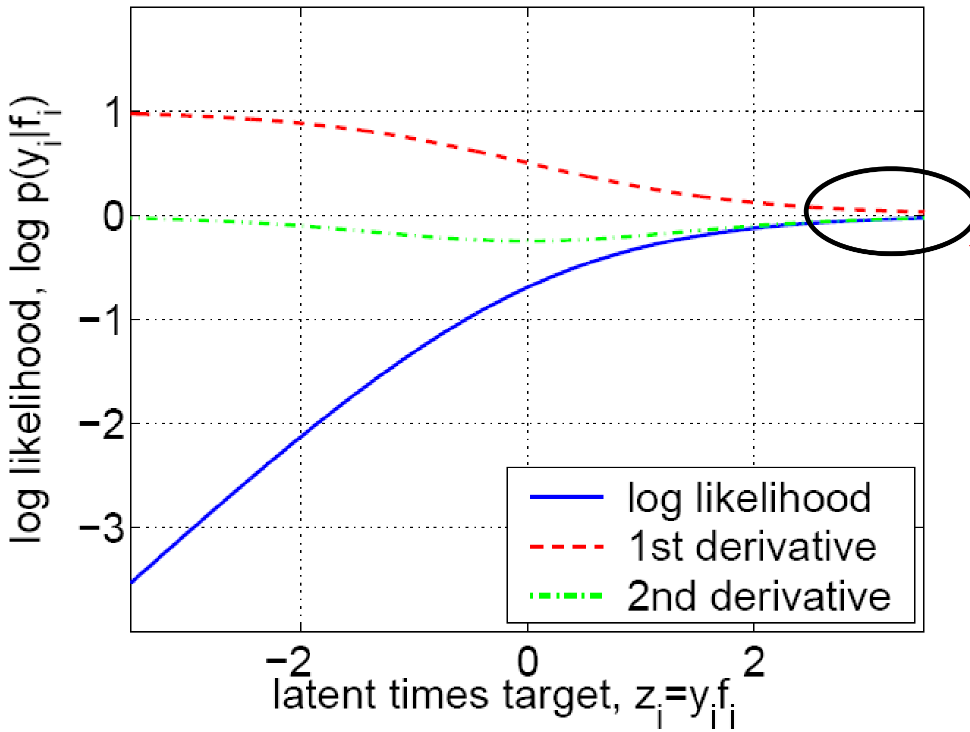
$$W_{ii} = \nabla^2_{f_i} \log p(y_i|f_i) = -\pi_i(1 - \pi_i)$$

where $\pi_i = p(y_i = 1|f_i)$ and $t_i = (y_i + 1)/2$.

The Hessian is negative definite $\Rightarrow$ **The objective is concave and has unique maxima**

Laplace Approximation for GP

- Logistic regression likelihood



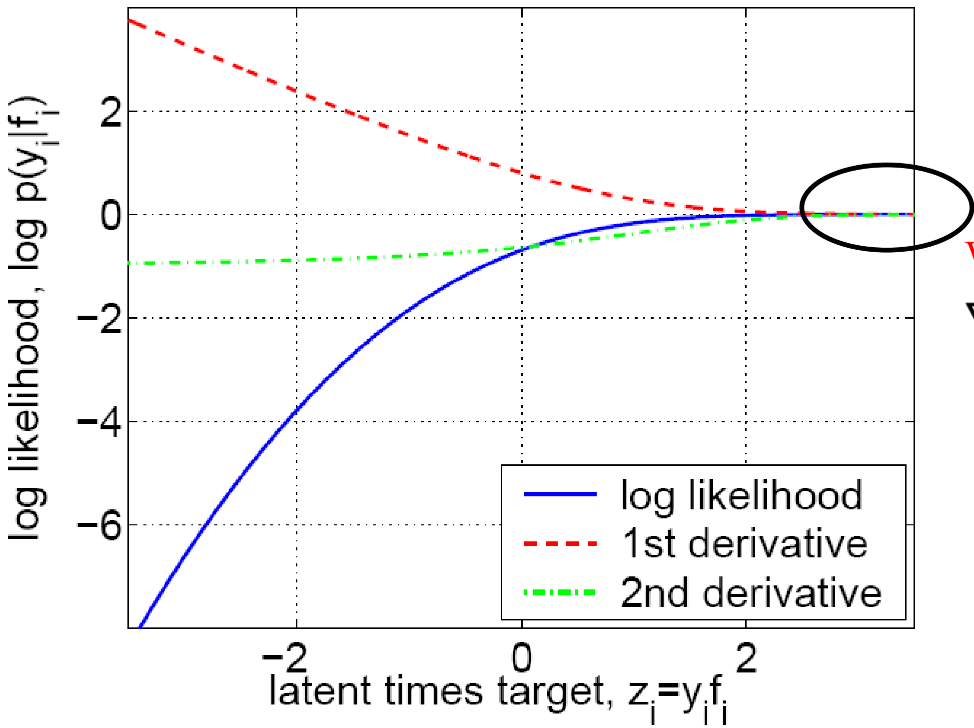
Well-explained Region

$$\nabla_{f_i} \log p(y_i|f_i) \approx 0$$

- How about negative examples?

Laplace Approximation for GP

- Probit regression likelihood



Well-explained Region

$$\nabla_{f_i} \log p(y_i|f_i) \approx 0$$

  - How about negative examples?

Laplace Approximation for GP

- The derivatives of the log posterior are

$$\nabla\psi(\mathbf{f}) = \nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f}$$

$$\nabla^2\psi(\mathbf{f}) = \nabla^2 \log p(\mathbf{y}|\mathbf{f}) - K^{-1} = -W - K^{-1}$$

  - W is diagonal since data points are independent

- Finding the mode
  - *Existence of maximum*
  - At the maximum, we have $\nabla\psi(\mathbf{f}) = 0$

$$\hat{\mathbf{f}} = K\nabla \log p(\mathbf{y}|\hat{\mathbf{f}})$$

  - No-closed form solution, numerical methods are needed

$$\mathbf{f}^{t+1} = \mathbf{f}^t - (\nabla^2\psi)^{-1}\nabla\psi = \mathbf{f}^t + (W + K^{-1})^{-1}(\nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f})$$

$$= (W + K^{-1})^{-1}(W\mathbf{f}^t + \nabla \log p(\mathbf{y}|\mathbf{f}))$$

Laplace Approximation for GP

- The derivatives of the log posterior are

$$\nabla \psi(\mathbf{f}) = \nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f}$$

$$\nabla^2 \psi(\mathbf{f}) = \nabla^2 \log p(\mathbf{y}|\mathbf{f}) - K^{-1} = -W - K^{-1}$$

  - W is diagonal since data points are independent

- Finding the mode
  - No-closed form solution, numerical methods are needed

$$\mathbf{f}^{t+1} = (W + K^{-1})^{-1}(W\mathbf{f}^t + \nabla \log p(\mathbf{y}|\mathbf{f}))$$

- The Gaussian approximation

$$q(\mathbf{f}|X, \mathbf{y}) = \mathcal{N}\left(\hat{\mathbf{f}}, (K^{-1} + W)^{-1}\right)$$

Laplace Approximation for GP

- Laplace approximation

$$q(\mathbf{f}|X,\mathbf{y}) = \mathcal{N}\Big(\hat{\mathbf{f}}, (K^{-1}+W)^{-1}\Big)$$

- Predictions as GP predictive mean

$$\mathbb{E}_q[f_*|X,\mathbf{y},\mathbf{x}_*] = \mathbf{k}(\mathbf{x}_*)^\top K^{-1}\hat{\mathbf{f}} = \mathbf{k}(\mathbf{x}_*)^\top \nabla \log p(\mathbf{y}|\hat{\mathbf{f}})$$

  - Positive examples have positive coefficients for their kernels

$$\nabla_{f_i} \log p(y_i = 1|f_i) = 1 - p(y_i = 1|f_i) > 0$$

  - Negative examples have negative coefficients for their kernels

$$\nabla_{f_i} \log p(y_i = -1|f_i) = -p(y_i = 1|f_i) < 0$$

  - Well-explained points don’t contribute strongly to predictions

$$\nabla_{f_i} \log p(y_i|f_i) \approx 0$$

Non-support vectors

Laplace Approximation for GP

- Laplace approximation

$$q(\mathbf{f}|X, \mathbf{y}) = \mathcal{N}\left(\hat{\mathbf{f}}, (K^{-1} + W)^{-1}\right)$$

- Predictions as GP predictive mean

$$\mathbb{E}_q[f_*|X, \mathbf{y}, \mathbf{x}_*] = \mathbf{k}(\mathbf{x}_*)^\top K^{-1}\hat{\mathbf{f}} = \mathbf{k}(\mathbf{x}_*)^\top \nabla \log p(\mathbf{y}|\hat{\mathbf{f}})$$

  - Then, the response variable is predicted as (MAP prediction)

$$\hat{y}_* = \sigma(\mathbb{E}_q[f_*|X, \mathbf{y}, \mathbf{x}_*])$$

  - Alternative average prediction

$$\hat{y}_* \approx \int \sigma(f_*) q(f_*|X, \mathbf{y}, \mathbf{x}_*) df_*$$

Weakness of Laplace Approximation

- Directly only applicable to real-valued variables
  - Based on Gaussian distribution
- May be applicable to transformed variable
  - If $0 < \tau < \infty$, then consider Laplace approximation of $\ln \tau$
- Based purely on a specific value of the variable
  - Expansion on local maxima

GPs for Multi-class Classification

- Latent functions for n training points and for C classes

$$\mathbf{f} = (f_1^1, \ldots, f_n^1, f_1^2, \ldots, f_n^2, \ldots, f_1^C, \ldots, f_n^C)^\top$$

- Using multiple independent GPs, one for each category

$$\forall c \in \mathcal{C} : \quad f^c(\mathbf{x}) \sim \mathcal{GP}(m^c(\mathbf{x}), \kappa^c(\mathbf{x}, \mathbf{x}'))$$

- Using softmax function to get the class probability

$$p(y_i^c | \mathbf{f}_i) = \frac{\exp(f_i^c)}{\sum_{c'} \exp(f_i^{c'})}$$

Notation: $\mathbf{y} = (y_1^1, \ldots, y_n^1, y_1^2, \ldots, y_n^2, \ldots, y_1^C, \ldots, y_n^C)^\top$

$\forall i :$ only one of y_i^c is 1. all other $C - 1$ entries are 0.

Laplace Approximation for Multi-class GP

- The log of un-normalized posterior is

$$\psi(\mathbf{f}) = \mathbf{y}^\top \mathbf{f} - \sum_n \log(\sum_c \exp f_i^c) - \frac{1}{2}\mathbf{f}^\top K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{Cn}{2}\log 2\pi$$

- We have $\nabla\psi(\mathbf{f}) = -K^{-1}\mathbf{f} + \mathbf{y} - \pi, \text{ where } \pi_i^c = p(y_i^c|\mathbf{f}_i)$

$$\nabla^2\psi(\mathbf{f}) = -K^{-1} - W, \text{ where } W = \mathrm{diag}(\pi) - \Pi\Pi^\top$$

- Then, the mode is

$$\hat{\mathbf{f}} = K(\mathbf{y} - \hat{\pi})$$

  - Newton method can be applied with the above Hessian

Uncorrelated processes between classes:

$$K = \begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & K_C \end{bmatrix} \qquad \Pi = \begin{bmatrix} \mathrm{diag}(\pi^1) \\ \mathrm{diag}(\pi^2) \\ \vdots \\ \mathrm{diag}(\pi^C) \end{bmatrix}$$

Laplace Approximation for Multi-class GP

- Predictions with the Gaussian approximation

$$\hat{\mathbf{f}} = K(\mathbf{y} - \hat{\pi})$$

$$q(\mathbf{f}|X, \mathbf{y}) = \mathcal{N}(\hat{\mathbf{f}}, (W + K^{-1})^{-1})$$

  - The predictive mean for class c is

$$q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*) = \int p(\mathbf{f}_*|X, \mathbf{x}_*, \mathbf{f}) q(\mathbf{f}|X, \mathbf{y}) d\mathbf{f}$$

    - which is Gaussian as both terms in the product are Gaussian
    - the mean and co-variance are

$$\mathbb{E}_q[f^c(\mathbf{x}_*)|X, \mathbf{y}, \mathbf{x}_*] = \mathbf{k}_c(\mathbf{x}_*)^\top K_c^{-1} \hat{\mathbf{f}}^c = \mathbf{k}_c(\mathbf{x}_*)^\top (\mathbf{y}^c - \hat{\pi}^c)$$

$$\mathrm{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*) = \mathrm{diag}(\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)) - Q_*^\top (K + W^{-1})^{-1} Q_*$$

$$Q_* = \begin{bmatrix} \mathbf{k}_1(\mathbf{x}_*) & 0 & 0 & 0 \\ 0 & \mathbf{k}_2(\mathbf{x}_*) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{k}_C(\mathbf{x}_*) \end{bmatrix}$$

Covariance Functions

- The only requirement for covariance matrix is the positive semidefinite
- Many covariance functions, hyper-parameters make influence

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \Big(\frac{\sqrt{2\nu}}{\ell} r\Big)^{\nu} K_{\nu} \Big(\frac{\sqrt{2\nu}}{\ell} r\Big)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\big(-(\frac{r}{\ell})^{\gamma}\big)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\Big(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\Big)$		$\surd$

S: stationary; ND: non-degenerate. Degenerate covariance functions have finite rank

Covariance Functions

- Squared Exponential Kernel

$$k(x_p, x_q) = \sigma_f^2 \exp\left[\frac{-(x_p - x_q)^2}{2l^2}\right]$$

  - Infinitely differentiable
  - Equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, not just training points!
- Gaussian-shaped basis functions

$$\forall c \in [c_{min}, c_{max}] : \ \phi_c(x) = \exp(-\frac{(x-c)^2}{2l^2})$$

  - For the finite case, let the prior $\mathbf{w} \sim \mathcal{N}(0, \sigma_p^2 I)$, we have a GP with covariance function

$$\kappa(x_p, x_q) = \sigma_p^2 \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q)$$

  - For the infinite limit, we can show

$$\frac{\sigma_p^2}{\Delta H} \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q) \xrightarrow{N \to \infty} \sqrt{\pi} l \sigma_p^2 \exp(-\frac{(x_p - x_q)^2}{2(\sqrt{2}l)^2}) \qquad \Delta H = \frac{N}{c_{\max} - c_{\min}}$$

basis functions per unit interval.

Covariance Functions

- Squared Exponential Kernel $\Delta H = \frac{N}{c_{\max} - c_{\min}}$

$$\frac{\sigma_p^2}{\Delta H} \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q) \xrightarrow{N\to\infty} \sqrt{\pi} l \sigma_p^2 \exp\Big(- \frac{(x_p - x_q)^2}{2(\sqrt{2}l)^2} \Big)$$

  - Proof: (a set of uniformly distributed basis functions)

$$\forall c \in [c_{min}, c_{max}]: \ \ \phi_c(x) = \exp(-\frac{(x-c)^2}{2l^2})$$

$$\lim_{N\to\infty} \frac{\sigma_p^2}{\Delta H} \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q) = \sigma_p^2 \int_{c_{\min}}^{c_{\max}} \phi_c(x_p)\phi_c(x_q) dc$$

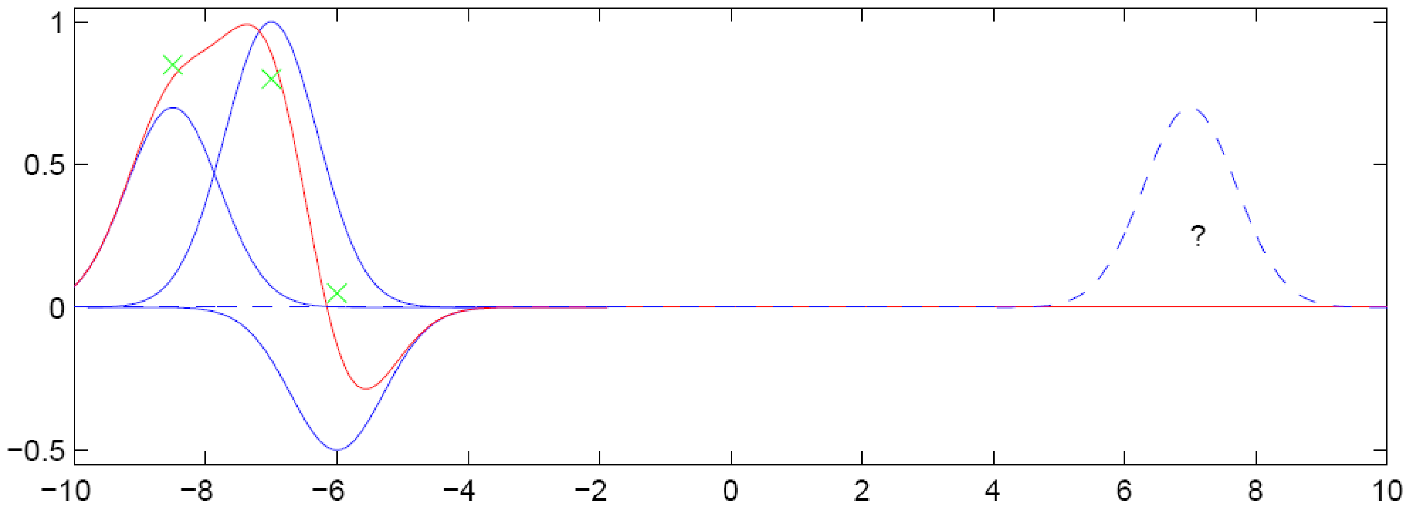
    - Let the integral interval go to infinity, we get

$$\kappa(x_p, x_q) = \sigma_p^2 \int_{-\infty}^{\infty} \exp\Big(- \frac{(x_p - c)^2}{2l^2} \Big) \exp\Big(- \frac{(x_q - c)^2}{2l^2} \Big) dc$$

$$= \sqrt{\pi} l \sigma_p^2 \exp\Big(- \frac{(x_p - x_q)^2}{2(\sqrt{2}l)^2} \Big)$$

Using finitely many basis functions can be dangerous!

- Missed components



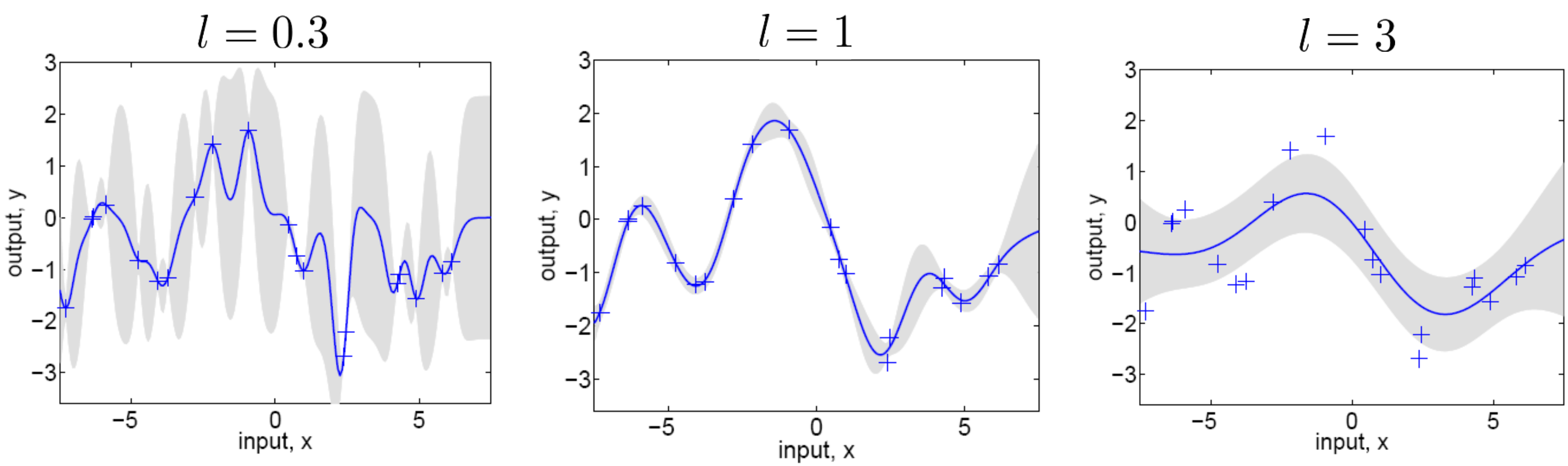
- Not full rank

Adaptation of Hyperparameters

- Characteristic length scale parameter l

$$k(x_p, x_q) = \sigma_f^2 \exp\left[\frac{-(x_p - x_q)^2}{2l^2}\right] + \sigma_n^2 \delta p, q$$

  - Roughly measures how far we need to go in order to make the data points un-related (or the function value change significantly)
  - Larger l gives smoother functions (i.e., simpler functions)



Adaptation of Hyperparameters

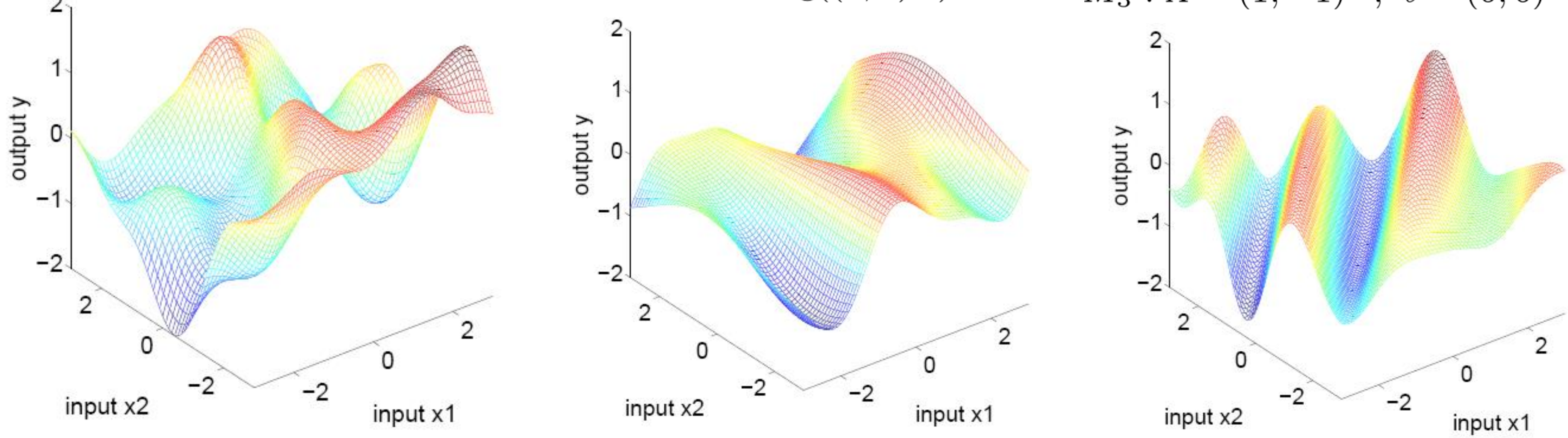
- Squared exponential covariance function

$$\kappa(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\Big(-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)^\top M (\mathbf{x}_p - \mathbf{x}_q)\Big) + \sigma_n^2 \delta_{p,q}$$

  - Hyper-parameters $\theta = (M, \sigma_f^2, \sigma_n^2)$
  - Possible choices of M

$$M_1 = \ell^{-2} I, \quad M_2 = \mathrm{diag}(\ell)^{-2}, \quad M_3 = \Lambda\Lambda^\top + \mathrm{diag}(\ell)^{-2}$$

$M_1 = I$ $\qquad$ $M_2 = \mathrm{diag}((1,3)^\top)^{-2}$ $\qquad$ $M_3 : \Lambda = (1,-1)^\top,\ \ell = (6,6)^\top$



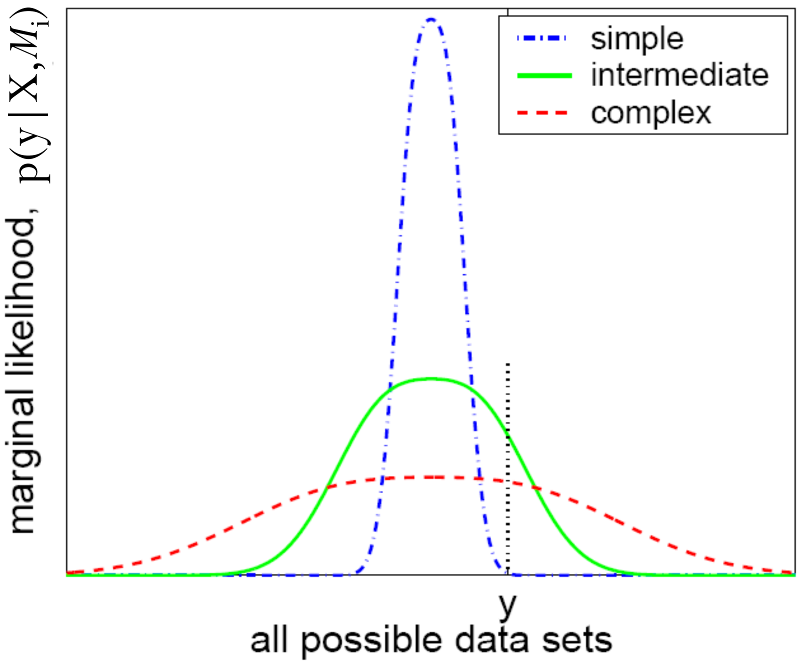
Marginal Likelihood for Model Selection

- A Bayesian approach to model selection
  - Let $\mathcal{M}_i$ denote a family of models. Each $\mathcal{M}_i$ is characterized by some parameters θ
  - The marginal likelihood (evidence) is

$$p(\mathbf{y}|X, \mathcal{M}_i) = \int \underbrace{p(\mathbf{y}|X, \theta, \mathcal{M}_i)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{M}_i)}_{\text{prior}} d\theta$$

  - An automatic trade-off between data fit and model complexity (see next slide …)

Marginal Likelihood for Model Selection



- Simple models account for a limited range of data sets; complex models account for a broad range of data sets.
- For a particular data set y, the margin likelihood prefers a model of intermediate complexity over too simple or too complex ones

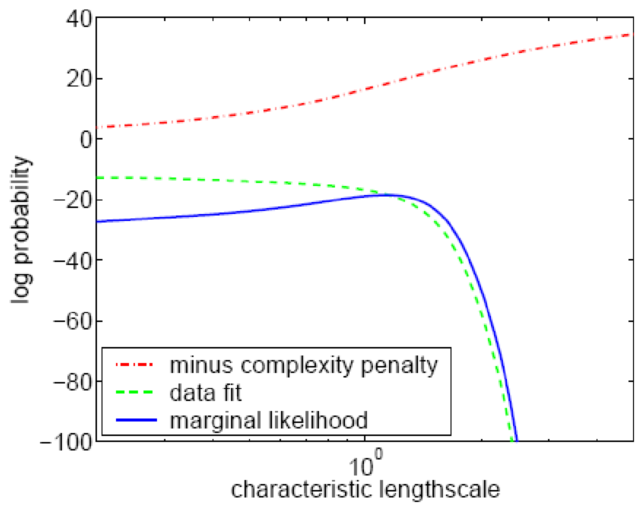
Marginal Likelihood for GP

- Marginal likelihood can be used to estimate the hyper-parameters for GP
- For GP regression, we have

$$\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi$$

data fit **model complexity**

where $K_y = K_f + \sigma^2 I$ for noisy targets $\mathbf{y}$.



Marginal Likelihood for GP

- Marginal likelihood can be used to estimate the hyper-parameters for GP
- For GP regression, we have

$$\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi$$

where $K_y = K_f + \sigma^2 I$ for noisy targets $\mathbf{y}$.

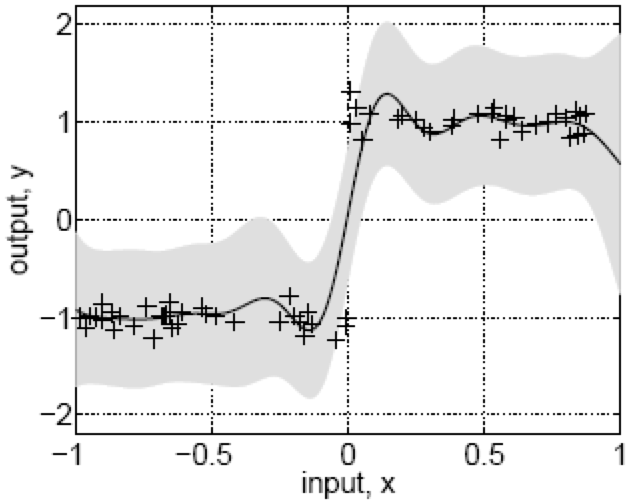
- Then, we can do gradient descent to solve

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{y}|X,\theta)$$

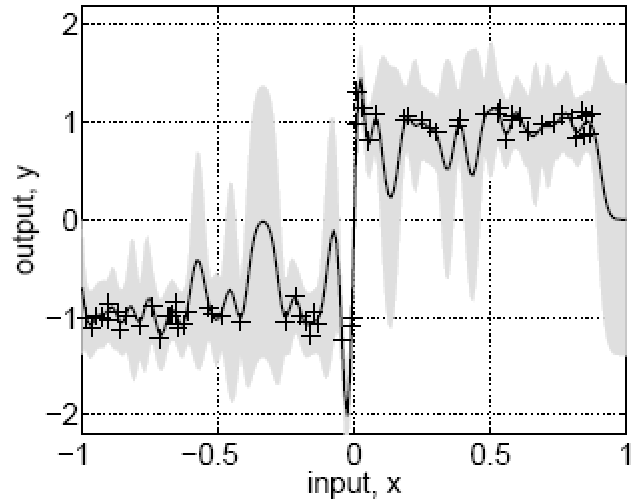
- For GP classification, we need Laplace approximation to compute the marginal likelihood.

Other Model Selection Methods

- When the number of parameters is small, we can do
  - K-fold cross-validation (CV)
  - Leave-one-out cross-validation (LOO-CV)
- Different selection methods usually lead to different results



Marginal likelihood estimation



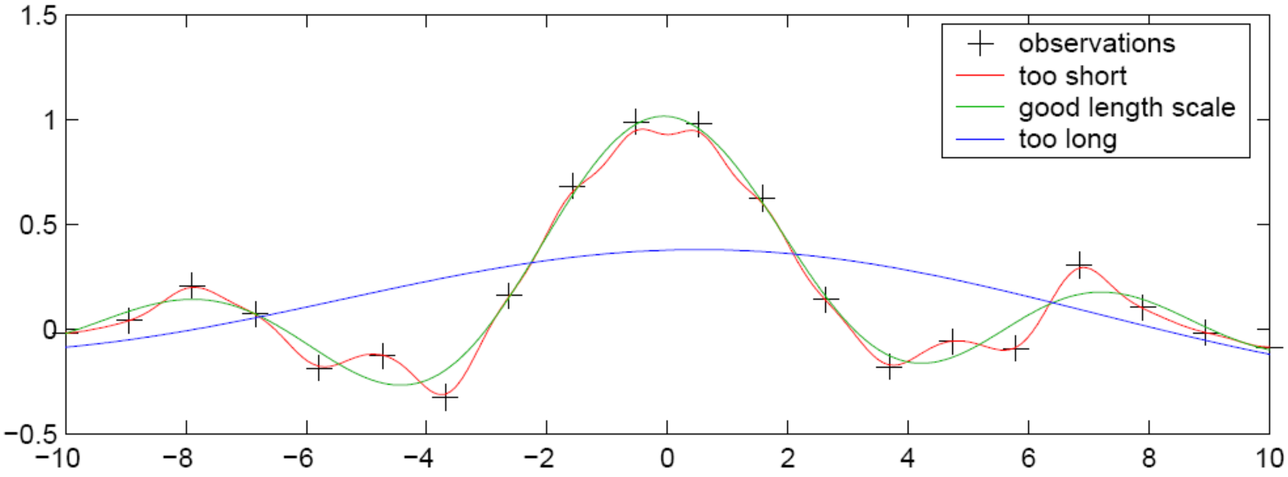
LOO-CV

Hyperparameters of Covariance Function

- Squared Exponential

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right]$$

  - Hyperparameters: maximum allowable covariance, and Length parameter



– The mean posterior predictive functions for three different length-scales
– Green one learned by maximum marginal likelihood
– Too short one can almost exactly fits the data!

Other Inference Methods

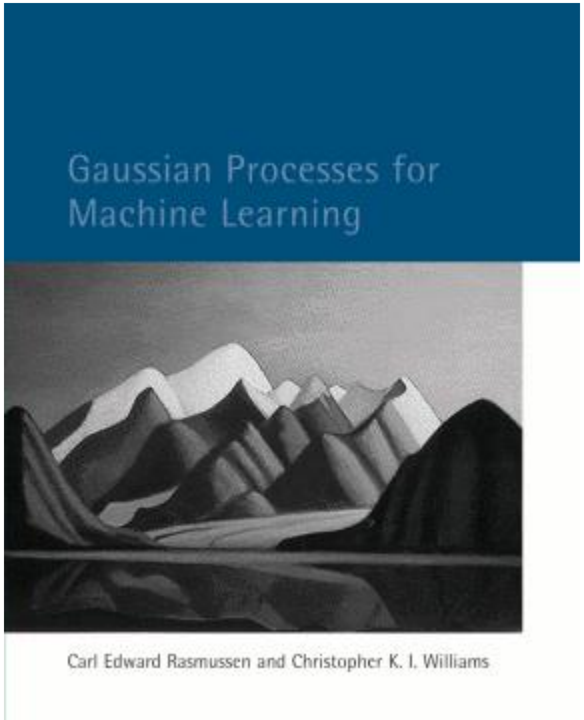
- Markov Chain Monte Carlo methods
- Expectation Propagation
- Variational Approximation

Other Issues

- Multiple outputs
- Noise models with correlations
- Non-Guassian likelihood
- Mixture of GPs
- Student's t process
- Latent variable models
- …

References

- Rasmussen & Williams. Gaussian Process for Machine Learning, 2006.



- The Gaussian Process website: http://www.gaussianprocess.org/

Source Code

- GPStuff
- http://becs.aalto.fi/en/research/bayes/gpstuff/