SC4/SM8 Advanced Topics in Statistical Machine Learning

Gaussian Processes

Dino Sejdinovic
Department of Statistics
Oxford

Slides and other materials available at:
http://www.stats.ox.ac.uk/~sejdinov/atsml/

Parametric vs Nonparametric models

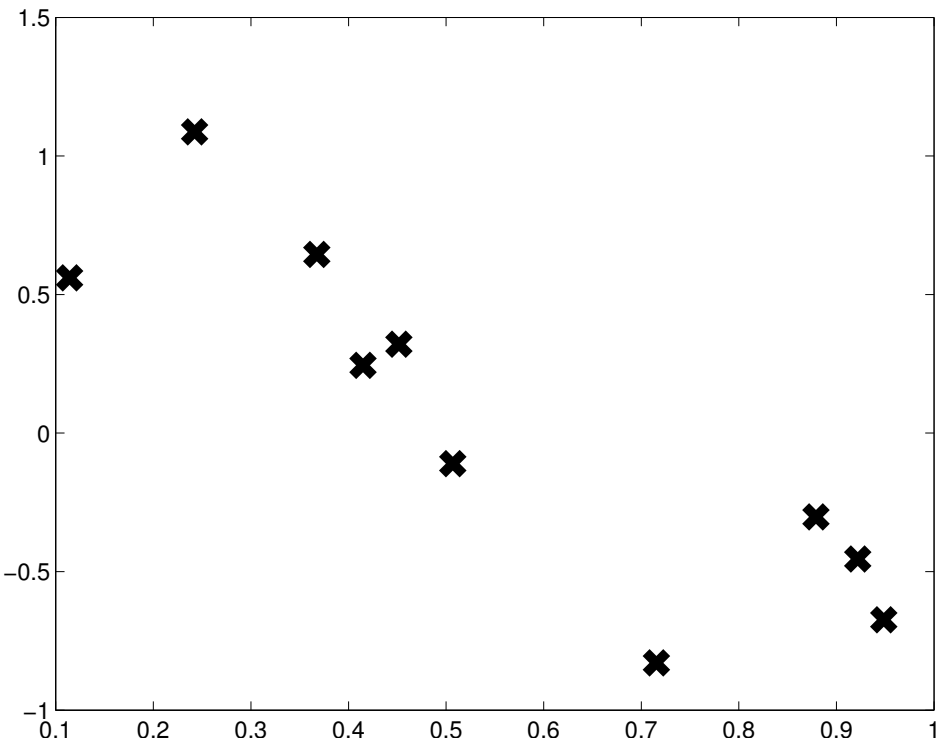
- **Parametric models** have a fixed finite number of parameters, regardless of the dataset size. In the Bayesian setting, given the parameter vector θ, the predictions are independent of the data $\mathcal{D}$.

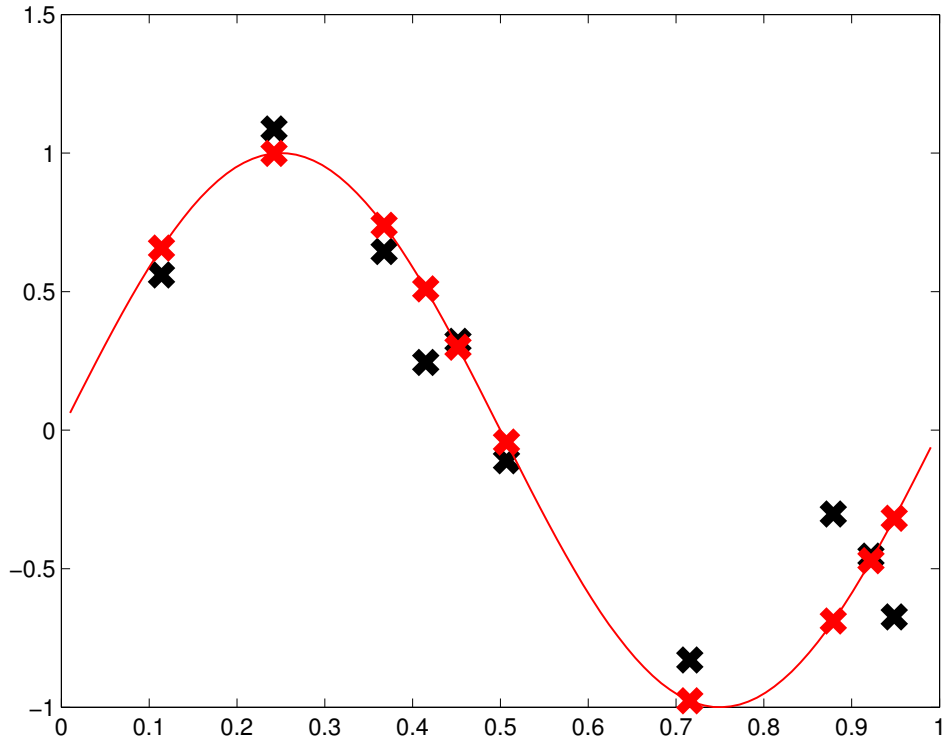
$$p(\tilde{x}, \theta|\mathcal{D}) = p(\theta|\mathcal{D})p(\tilde{x}|\theta)$$

  Parameters can be thought of as a data summary: communication channel flows from data to the predictions through the parameters.

- **Nonparametric models** allow the number of "parameters" to grow with the dataset size. Alternatively, predictions depend on the data (and the hyperparameters).

Regression





- We are given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Regression: learn the underlying real-valued function $f(x)$.

Different Flavours of Regression

- We can model response y_i as a noisy version of the underlying function f evaluated at input x_i:

$$y_i | f(x_i) \sim \mathcal{N}(f(x_i), \sigma^2)$$

  Appropriate loss: $L(y, f(x)) = (y - f(x))^2$

- **Frequentist Parametric** approach: model f as f_θ for some parameter vector θ. Fit θ by ML / ERM with squared loss (linear regression).
- **Frequentist Nonparametric** approach: model f as the unknown parameter taking values in an infinite-dimensional space of functions. Fit f by regularized ML / ERM with squared loss (kernel ridge regression)
- **Bayesian Parametric** approach: model f as f_θ for some parameter vector θ. Put a prior on θ and compute a posterior $p(\theta|\mathcal{D})$ (Bayesian linear regression).
- **Bayesian Nonparametric** approach: treat f as the random variable taking values in an infinite-dimensional space of functions. Put a prior over functions $f \in \mathcal{F}$, and compute a posterior $p(f|\mathcal{D})$ (Gaussian Process regression).

- Just work with the function values at the inputs $\mathbf{f} = (f(x_1), \ldots, f(x_n))^\top$
- What properties of the function can we incorporate?
  - Multivariate normal prior on $\mathbf{f}$:

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$

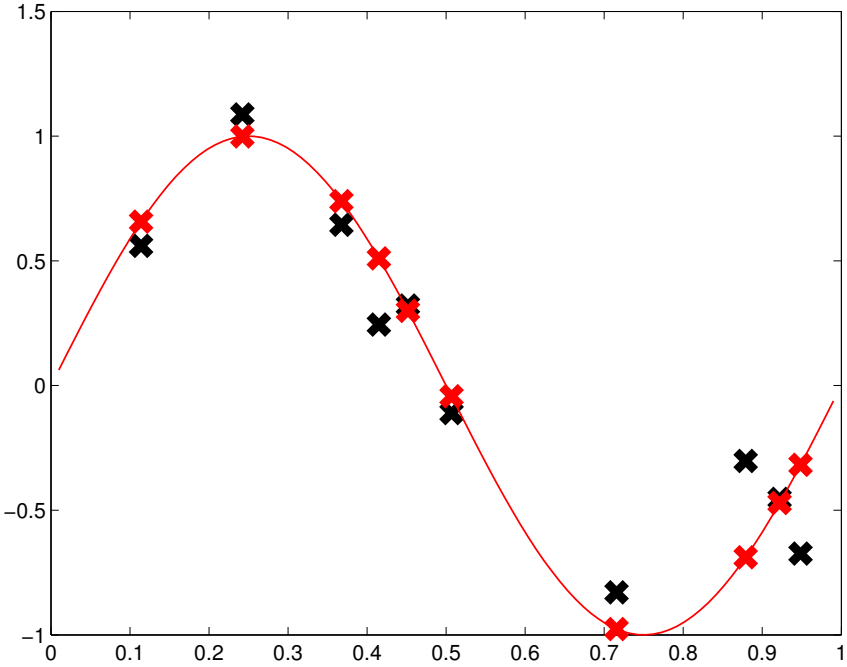
  - Use a kernel function k to define $\mathbf{K}$:

$$\mathbf{K}_{ij} = k(x_i, x_j)$$

  - Expect regression functions to be smooth: If x and x' are close by, then $f(x)$ and $f(x')$ have similar values, i.e. strongly correlated.

$$\begin{pmatrix} f(x) \\ f(x') \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} k(x,x) & k(x,x') \\ k(x',x) & k(x',x') \end{pmatrix} \right)$$

The prior $p(\mathbf{f})$ encodes our prior knowledge about the function.



- Model:

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$

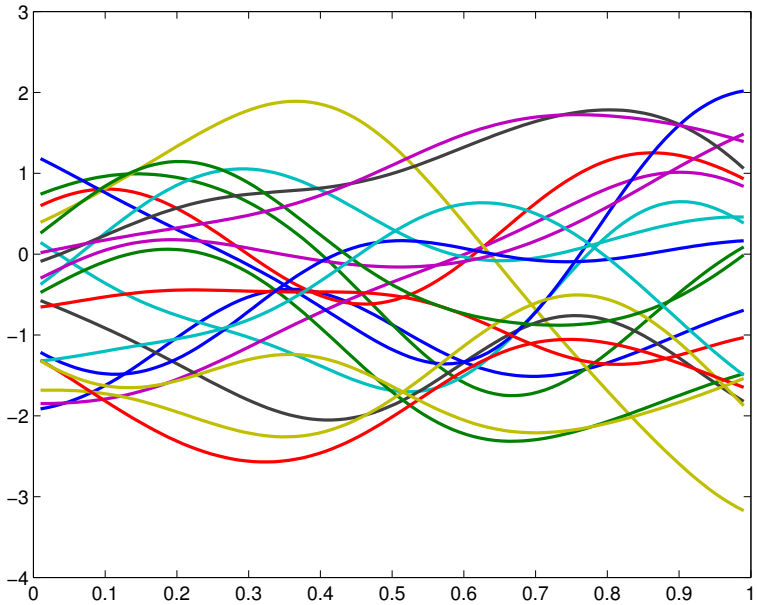
$$y_i | f_i \sim \mathcal{N}(f_i, \sigma^2)$$

Gaussian Processes

- What does a multivariate normal prior mean?
- Imagine $\mathbf{x}$ forms an infinitesimally dense grid of data space. Simulate prior draws

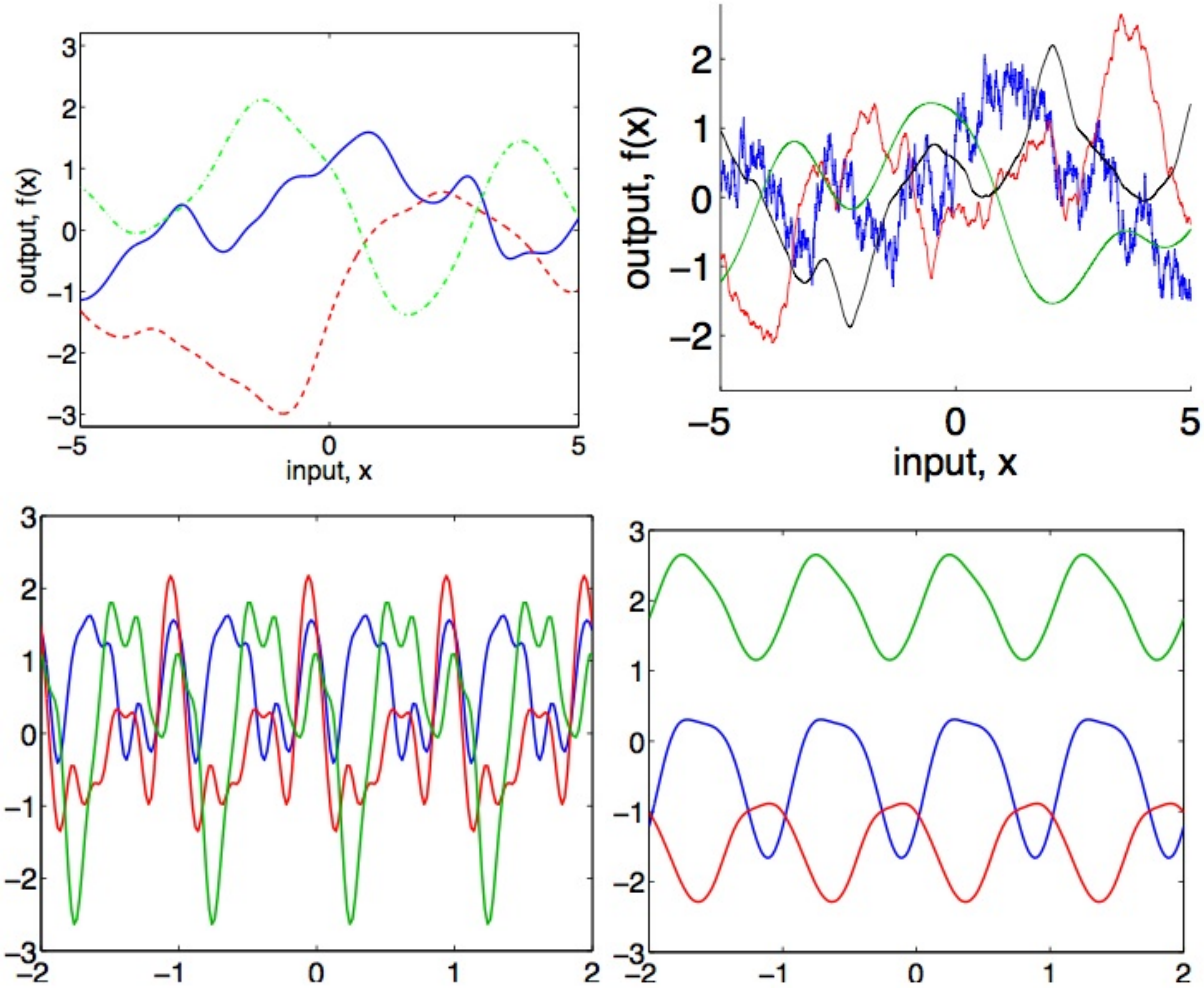
$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$

  Plot f_i vs x_i for $i = 1, \ldots, n$.
- The corresponding prior over functions is called a **Gaussian Process** (GP): any finite number of evaluations of which follow a Gaussian distribution.



Gaussian Processes

- Different kernels lead to different function characteristics.



Carl Rasmussen. Tutorial on Gaussian Processes at NIPS 2006.

Gaussian Processes

$$\mathbf{f}|\mathbf{x} \sim \mathcal{N}(0, \mathbf{K})$$
$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

- Posterior distribution:

$$\mathbf{f}|\mathbf{y} \sim \mathcal{N}(\mathbf{K}(\mathbf{K}+\sigma^2 I)^{-1}\mathbf{y}, \mathbf{K} - \mathbf{K}(\mathbf{K}+\sigma^2 I)^{-1}\mathbf{K})$$

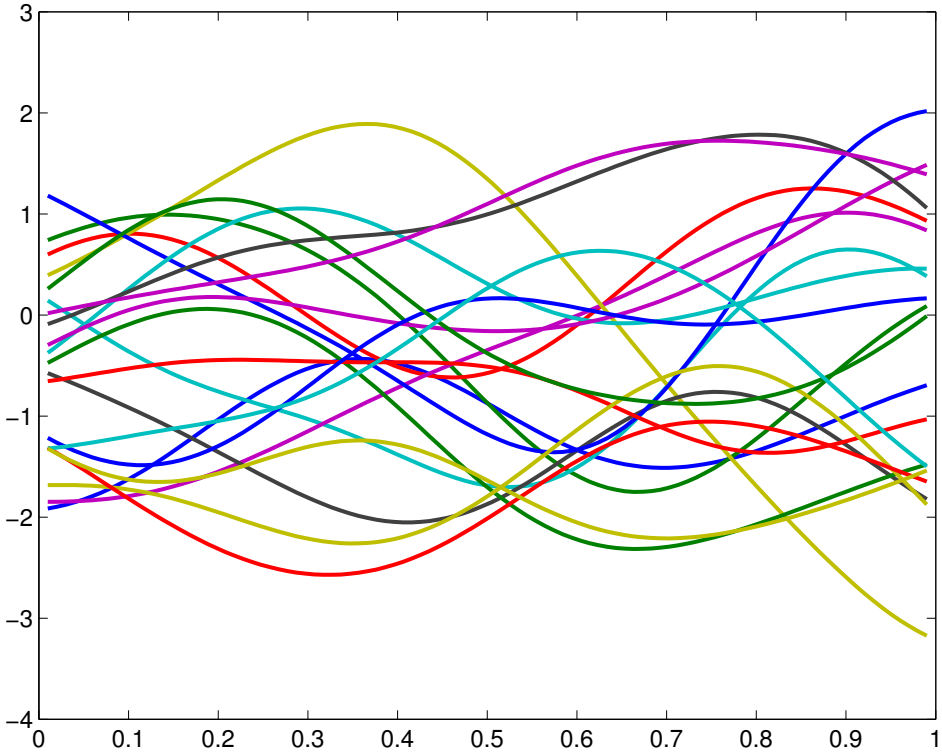
- Posterior predictive distribution: Suppose $\mathbf{x}'$ is a test set. We can extend our model to include the function values $\mathbf{f}'$ at the test set:

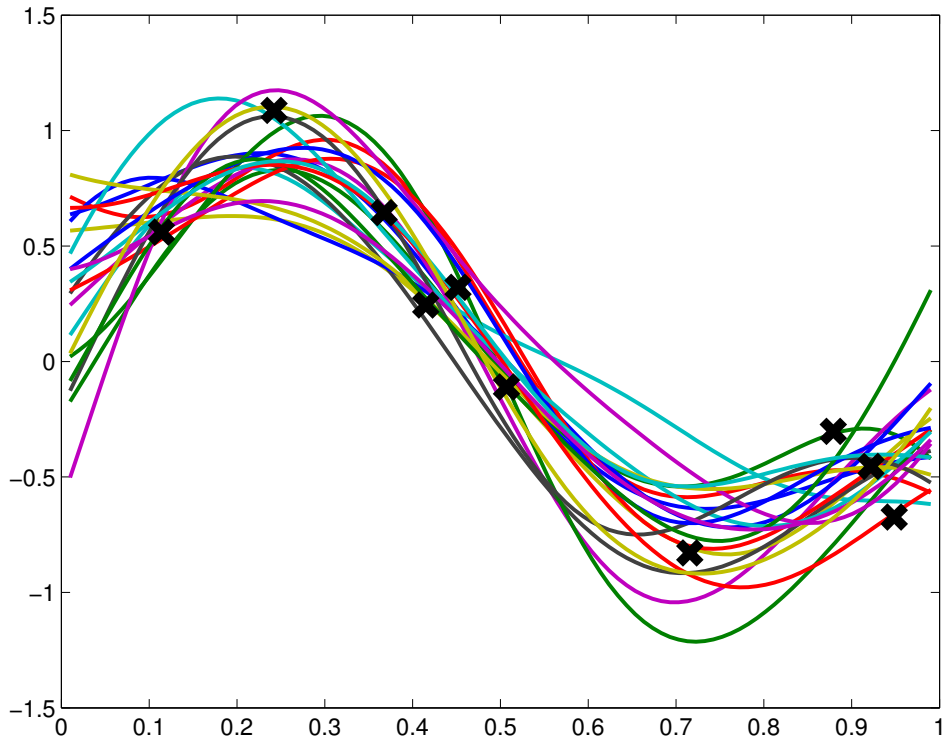
$$\begin{pmatrix}\mathbf{f}\\ \mathbf{f}'\end{pmatrix}|\mathbf{x}, \mathbf{x}' \sim \mathcal{N}\left(\begin{pmatrix}0\\ 0\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{\mathbf{xx}} & \mathbf{K}_{\mathbf{xx}'}\\ \mathbf{K}_{\mathbf{x}'\mathbf{x}} & \mathbf{K}_{\mathbf{x}'\mathbf{x}'}\end{pmatrix}\right)$$
$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

  where $\mathbf{K}_{\mathbf{xx}'}$ is matrix with (i,j)-th entry $k(x_i, x'_j)$.
- Some manipulation of multivariate normals gives:

$$\mathbf{f}'|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}}+\sigma^2 I)^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}}+\sigma^2 I)^{-1}\mathbf{K}_{\mathbf{xx}'}\right)$$

Gaussian Processes





GP regression demo: http://www.tmpl.fi/gp/

Hyperparameters: Maximum marginal likelihood

Marginal likelihood of the hyperparameter vector $\theta = (\nu, \sigma^2)$ (ν: kernel parameters, σ^2: noise in the observation model)

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f}, \theta) p(\mathbf{f}|\theta) d\mathbf{f} = \mathcal{N}\left(\mathbf{y}; 0, \mathbf{K}_\nu + \sigma^2 I\right).$$

Writing $\mathbf{K}_{\theta+} = \mathbf{K}_\nu + \sigma^2 I$, marginal log-likelihood is

$$\log p(\mathbf{y}|\theta) = -\frac{1}{2} \log |\mathbf{K}_{\theta+}| - \frac{1}{2} \mathbf{y}^\top \mathbf{K}_{\theta+}^{-1} \mathbf{y} - \frac{n}{2} \log(2\pi). \tag{1}$$

Typically a nonconvex function of θ.

Hyperparameters: Bayesian treatment

Place a prior $p(\theta)$ on θ and draw samples $\{\theta_j\}$ from the posterior

$$p(\theta|\mathbf{y}) \quad \propto \quad p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{y}|\mathbf{f},\theta)p(\mathbf{f}|\theta)d\mathbf{f}.$$

Integrate uncertainty over hyperparameters into predictions:

$$\begin{aligned} p(\mathbf{f}'|\mathbf{y}) &= \int p(\mathbf{f}'|\mathbf{y},\theta)p(\theta|\mathbf{y})d\theta \\ &\approx \sum_j p(\mathbf{f}'|\mathbf{y},\theta_j). \end{aligned}$$

GP with a logistic link

Consider the binary classification model with classes -1 and $+1$. Need to map Gaussian process into $(0, 1)$ with a nonlinear activation/link function, e.g.

$$p(y_i = +1|f(x_i)) = \sigma(f(x_i)) = \frac{1}{1 + e^{-f(x_i)}}. \tag{2}$$

Non-conjugate so exact posterior inference intractable.

Laplace approximation

Find MAP $\hat{\mathbf{f}}^{\text{MAP}}$ by maximizing

$$\begin{aligned}\log p(\mathbf{f}|\mathbf{y}) &= \text{const} + \log p(\mathbf{f}) + \log p(\mathbf{y}|\mathbf{f}) \\ &= \text{const} - \frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} + \sum_{i=1}^{n} \log \sigma(y_i f(x_i)).\end{aligned}$$

Gradient:

$$\frac{\partial \log p(\mathbf{f}|\mathbf{y})}{\partial \mathbf{f}} = -\mathbf{K}^{-1}\mathbf{f} + \mathbf{g}_\mathbf{f}$$

where the gradient of the likelihood is $\mathbf{g}_\mathbf{f} = \frac{\partial \log p(\mathbf{y}|\mathbf{f})}{\partial \mathbf{f}}$ with $[\mathbf{g}_\mathbf{f}]_i = \frac{\partial \log p(\mathbf{y}|\mathbf{f})}{\partial f_i} = \sigma(-y_i f(x_i))y_i$.

Laplace approximation

Hessian:

$$\frac{\partial^2 \log p(\mathbf{f}|\mathbf{y})}{\partial \mathbf{f} \partial \mathbf{f}^\top} = -\mathbf{K}^{-1} - \mathbf{D}_\mathbf{f},$$

where $\mathbf{D}_\mathbf{f} = -\frac{\partial^2 \log p(\mathbf{y}|\mathbf{f})}{\partial \mathbf{f} \partial \mathbf{f}^\top}$ is the negative Hessian of the log-likelihood, which is an $n \times n$ diagonal matrix, $(\mathbf{D}_\mathbf{f})_{ii} = \sigma(f(x_i))\sigma(-f(x_i)) \geq 0$.
Approximation to the posterior of $\mathbf{f}$:

$$\tilde{p}(\mathbf{f}|\mathbf{y}) = \mathcal{N}\left(\mathbf{f} \middle| \hat{\mathbf{f}}^{\text{MAP}}, \left(\mathbf{K}^{-1} + \mathbf{D}_{\hat{\mathbf{f}}^{\text{MAP}}}\right)^{-1}\right).$$

Approximation to the predictive posterior:

$$\tilde{p}(\mathbf{f}'|\mathbf{y}) = \mathcal{N}\left(\mathbf{f}' \mid \mathbf{K}_{\mathbf{x}'\mathbf{x}}\mathbf{K}_{\mathbf{x}\mathbf{x}}^{-1}\hat{\mathbf{f}}^{\text{MAP}}, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}\left(\mathbf{K}_{\mathbf{x}\mathbf{x}} + \mathbf{D}_{\hat{\mathbf{f}}^{\text{MAP}}}^{-1}\right)^{-1}\mathbf{K}_{\mathbf{x}\mathbf{x}'}\right). \quad (3)$$

Same mean as the plug-in predictive $p\left(\mathbf{f}'|\hat{\mathbf{f}}^{\text{MAP}}\right)$ but the plug-in underestimates the variance.

Probit model

Can use probit instead of logistic, i.e.

$$p(y_i = +1|f(x_i)) = \Phi(f(x_i)), \tag{4}$$

where $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} dt$ is the standard normal cdf.
Analogous derivations by considering the gradient and Hessian of the log-posterior

$$\log p(\mathbf{f}|\mathbf{y}) \quad = \quad \text{const} - \frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} + \sum_{i=1}^{n} \log \Phi(y_i f(x_i)).$$

It suffices to replace

$$\begin{aligned} (\mathbf{g_f})_i &= \frac{y_i \phi(f_i)}{\Phi(y_i f_i)}, \\ (\mathbf{D_f})_{ii} &= \frac{\phi(f_i)^2}{\Phi(y_i f_i)^2} + \frac{y_i f_i \phi(f_i)}{\Phi(y_i f_i)} \end{aligned}$$

Large Scale Approximations

Kernel methods at scale

- Expressivity of kernel methods (rich, often infinite-dimensional hypothesis spaces) comes with a cost that scales at least quadratically in the number of observations n (due to needing to compute, store and often invert the Gram matrix)! We arrived at this by trying to avoid paying the cost in the dimension of the hypothesis space (e.g., for order d polynomial kernels, scales as $\binom{p+d}{d}$, and infinite for many kernels).
- But now we have to pay in terms of n which is problematic when we have a lot of observations (and this is exactly when we want to use a rich expressive model with a high-dimensional hypothesis class!)
- Scaling up kernel methods is a very active research area

  [Sonnenburg et al, 2006; Rahimi & Recht 2007; Le, Sarlos & Smola, 2013; Wilson et al, 2014; Dai et al, 2014; Sriperumbudur & Szabo, 2015].
- Main idea: study the desired hypothesis space and scale its dimension down - then undo the kernel trick!
- Errm... So we went the full circle (!?)
  explicit basis functions $\rightarrow$ implicit basis functions $\rightarrow$ explicit basis functions

Random Fourier features: Inverse Kernel Trick

Bochner's representation: any positive definite **translation-invariant** kernel on $\mathbb{R}^p$ can be written as

$$\begin{aligned} k(x,y) &= \int_{\mathbb{R}^p} \exp\left(i\omega^\top (x-y)\right) d\Lambda(\omega) \\ &= \int_{\mathbb{R}^p} \left\{\cos\left(\omega^\top x\right)\cos\left(\omega^\top y\right) + \sin\left(\omega^\top x\right)\sin\left(\omega^\top y\right)\right\} d\Lambda(\omega) \end{aligned}$$

for some positive measure (w.l.o.g. a probability distribution) Λ.

- Sample m frequencies $\{\omega_j\} \sim \Lambda$ and use a Monte Carlo estimator of the kernel function instead [Rahimi & Recht, 2007]:

$$\begin{aligned} \hat{k}(x,y) &= \frac{1}{m}\sum_{j=1}^{m} \left\{\cos\left(\omega_j^\top x\right)\cos\left(\omega_j^\top y\right) + \sin\left(\omega_j^\top x\right)\sin\left(\omega_j^\top y\right)\right\} \\ &= \langle \varphi_\omega(x), \varphi_\omega(y)\rangle_{\mathbb{R}^{2m}}, \end{aligned}$$

  with an explicit set of features $x \mapsto \frac{1}{\sqrt{m}}\left[\cos\left(\omega_1^\top x\right), \sin\left(\omega_1^\top x\right), \ldots\right]$.

- How fast does m need to grow with n? Sublinear for regression [Bach, 2015]

Inducing variables / Nyström

- Directly approximate the $n \times n$ Gram matrix K_{XX} of a set of inputs $\{x_i\}_{i=1}^n$ with

$$\hat{K}_{XX} = K_{XZ} K_{ZZ}^{-1} K_{ZX}$$

  where K_{ZZ} is $m \times m$ on "inducing" inputs $\{z_i\}_{i=1}^m$.
- Corresponds to explicit feature representation $x \mapsto K_{xZ} K_{ZZ}^{-1/2}$.
- Surrogate kernel $\hat{k}(x, x') = \langle k_|(\cdot, x), k_|(\cdot, x') \rangle$, where $k_|(\cdot, x)$ is a projection of $k(\cdot, x)$ to span $\{k(\cdot, z_1), \ldots, k(\cdot, z_m)\}$
- Often used in regression with Gaussian processes: with the use of Sherman-Morrison-Woodbury identity, reduces $O(n^3)$ cost to $O(nm^2)$.

  [Quiñonero-Candela and Rasmussen, 2005, Snelson and Ghahramani, 2006]
- m can grow much slower than n in regression without sacrificing performance [Rudi, Camoriano & Rosasco, 2015].