Probabilistic Models

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

Machine Learning

Outline

Outline

Predictions based on Probability

- Supervised learning, we are given a training set $\mathbb{X} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$
- Model $\mathbb{F}$: a collection of functions parametrized by Θ
- Goal: to train a function f such that, given a new data point $\boldsymbol{x}'$, the output value

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}'; \Theta)$$

  is closest to the correct label $\boldsymbol{y}'$
- Examples in $\mathbb{X}$ are usually assumed to be i.i.d. sampled from random variables $(\mathbf{x}, \mathbf{y})$ following some data generating distribution $\mathrm{P}(\mathbf{x}, \mathbf{y})$
- In probabilistic models, we write $f(\boldsymbol{x}'; \Theta)$ as $\mathrm{P}(\mathbf{y} = \boldsymbol{y} \,|\, \mathbf{x} = \boldsymbol{x}')$ and a prediction is made by:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\mathbf{y} = \boldsymbol{y} \,|\, \mathbf{x} = \boldsymbol{x}'; \Theta)$$

- How to find Θ?

Function (Θ) as Point Estimate

- Regard Θ (f) as an estimate of the "true" Θ^* (f^*)
  - Mapped from the training set $\mathbb{X}$
- ***Maximum a posteriori (MAP) estimation***:

$$\Theta_{\text{MAP}} = \arg\max_{\Theta} \mathrm{P}(\Theta \mid \mathbb{X}) = \arg\max_{\Theta} \frac{\mathrm{P}(\mathbb{X} \mid \Theta)\mathrm{P}(\Theta)}{\mathrm{P}(\mathbb{X})} = \arg\max_{\Theta} \mathrm{P}(\mathbb{X} \mid \Theta)\mathrm{P}(\Theta)$$

  - Θ is regarded as a random variable
- ***Maximum likelihood (ML) estimation***:

$$\Theta_{\text{ML}} = \arg\max_{\Theta} \mathrm{P}(\mathbb{X} \mid \Theta)$$

  - Assumes a uniform $\mathrm{P}(\Theta)$ (i.e., does not prefer a particular Θ)
- After being solved, $\Theta_{\text{ML/MAP}}$ is treated as a constant when make a prediction $\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}; \Theta_{\text{ML/MAP}})$

Outline

Outline

Probability Interpretation

- Assumption: $y = f^*(\mathbf{x}) + \varepsilon$, where
  - $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$
  - $f^*(\mathbf{x}; \boldsymbol{w}^*) = \boldsymbol{w}^{*\top}\mathbf{x}$ is a deterministic function
  - All variables in $\mathbf{x}$ are z-normalized, so there's no bias term b in f^*
- We have $(\mathrm{y} \,|\, \mathbf{x} = \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{w}^{*\top}\boldsymbol{x}, \beta^{-1})$
- So, out goal is to find $\boldsymbol{w}$ as close to $\boldsymbol{w}^*$ as possible such that:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\boldsymbol{y} \,|\, \mathbf{x} = \boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x}$$

  - Note that $\hat{y}$ is irrelevant to β, so we don't need to solve β
- ML estimation for $\boldsymbol{w}^*$:

$$\boldsymbol{w}_{\mathrm{ML}} = \arg\max_{\boldsymbol{w}} \mathrm{P}(\mathbb{X} \,|\, \boldsymbol{w})$$

ML Estimation I

- Problem:

$$\boldsymbol{w}_{\text{ML}} = \arg\max_{\boldsymbol{w}} \mathrm{P}(\mathbb{X} \,|\, \boldsymbol{w})$$

- Since we assume i.i.d. samples, we have

$$\begin{aligned}
\mathrm{P}(\mathbb{X} \,|\, \boldsymbol{w}) &= \textstyle\prod_{i=1}^{N} \mathrm{P}(\boldsymbol{x}^{(i)}, y^{(i)} \,|\, \boldsymbol{w}) = \prod_{i=1}^{N} \mathrm{P}(y^{(i)} \,|\, \boldsymbol{x}^{(i)}, \boldsymbol{w}) \mathrm{P}(\boldsymbol{x}^{(i)} \,|\, \boldsymbol{w}) \\
&= \textstyle\prod_{i=1}^{N} \mathrm{P}(y^{(i)} \,|\, \boldsymbol{x}^{(i)}, \boldsymbol{w}) \mathrm{P}(\boldsymbol{x}^{(i)}) = \prod_{i} \mathcal{N}(y^{(i)}; \boldsymbol{w}^{\top} \boldsymbol{x}^{(i)}, \sigma^2) \mathrm{P}(\boldsymbol{x}^{(i)}) \\
&= \textstyle\prod_{i} \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} (y^{(i)} - \boldsymbol{w}^{\top} \boldsymbol{x}^{(i)})^2 \right) \mathrm{P}(\boldsymbol{x}^{(i)})
\end{aligned}$$

- To make the problem tractable, we prefer "sums" over "products"
- We can instead maximize the ***log likelihood***

$$\begin{aligned}
&\arg\max_{\boldsymbol{w}} \log \mathrm{P}(\mathbb{X} \,|\, \boldsymbol{w}) \\
&= \arg\max_{\boldsymbol{w}} \log \left[\textstyle\prod_{i} \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} (y^{(i)} - \boldsymbol{w}^{\top} \boldsymbol{x}^{(i)})^2 \right) \mathrm{P}(\boldsymbol{x}^{(i)}) \right] \\
&= \arg\max_{\boldsymbol{w}} N \sqrt{\frac{\beta}{2\pi}} - \frac{\beta}{2} \textstyle\sum_{i} (y^{(i)} - \boldsymbol{w}^{\top} \boldsymbol{x}^{(i)})^2 + \sum_{i} \mathrm{P}(\boldsymbol{x}^{(i)})
\end{aligned}$$

- The optimal point does not change since log is monotone increasing

ML Estimation II

$$\arg\max_{\boldsymbol{w}} N\sqrt{\frac{\beta}{2\pi}} - \frac{\beta}{2}\sum_i (y^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)})^2 + \sum_i \mathrm{P}(\boldsymbol{x}^{(i)})$$

- Ignoring terms irrelevant to $\boldsymbol{w}$, we have

$$\boldsymbol{w}_{\mathrm{ML}} = \arg\min_{\boldsymbol{w}} \sum_i (y^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)})^2$$

- Effectively, we seek for $\boldsymbol{w}$ by ***minimizing the SSE*** (sum of square errors), as we have done before
  - Can solved analytically
  - Or numerically by, e.g., the stochastic gradient descent algorithm
- This new perspective explains our ad hoc choice of SSE for empirical risk minimization
  - Checking assumptions helps understand when model works the best
- Also motivates new models. Probabilistic model for classification?

Outline

Probabilistic Models for Binary Classification

- Probabilistic models:

$$\hat{y} = \arg\max_{y} \mathrm{P}(y \,|\, \boldsymbol{x}; \Theta)$$

- In regression, we assume $(\mathrm{y} \,|\, \boldsymbol{x}) \sim \mathcal{N}$ (based on $y = f^*(\boldsymbol{x}) + \varepsilon$)
- However, Gaussian distribution is ***not*** applicable to binary classification
  - The values of y should concentrate in either 1 or -1
- Which distribution to assume?
- Coin flipping: $(\mathrm{y} \,|\, \boldsymbol{x}) \sim \mathrm{Bernoulli}(\rho)$, where

$$\mathrm{P}(y \,|\, \boldsymbol{x}; \rho) = \rho^{y'} (1-\rho)^{(1-y')}, \text{ where } y' = \frac{y+1}{2} \in \{0, 1\}$$

- ML estimate $\mathrm{P}(\mathbb{X} \,|\, \rho)$? How to relate $\boldsymbol{x}$ to ρ?

Logistic Function

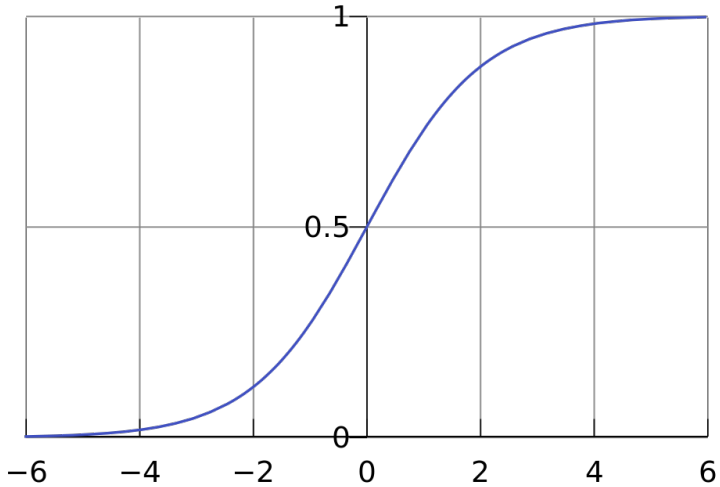
- Recall that the ***logistic function***

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

  is commonly used as a parametrizing function of the Bernoulli distribution

- We have

$$\mathrm{P}(y \,|\, \boldsymbol{x}; z) = \sigma(z)^{y'} \left(1 - \sigma(z)\right)^{(1-y')}$$



- The larger z, the higher chance we get a "positive flip"
- How to relate $\boldsymbol{x}$ to z?

Logistic Regression

- In ***logistic regression***, we let

$$z = \boldsymbol{w}^\top \boldsymbol{x}$$

  - Basically, z is the projection of $\boldsymbol{x}$ along the direction $\boldsymbol{w}$
- We have

$$\mathrm{P}(y \,|\, \boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^\top \boldsymbol{x})^{y'} [1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x})]^{(1-y')}$$

- Prediction:

$$\hat{y} = \arg\max_y \mathrm{P}(y \,|\, \boldsymbol{x}; \boldsymbol{w}) = \mathrm{sign}(\boldsymbol{w}^\top \boldsymbol{x})$$

- How to learn $\boldsymbol{w}$ from $\mathbb{X}$?
- ML estimation:

$$\boldsymbol{w}_{\mathrm{ML}} = \arg\max_{\boldsymbol{w}} \mathrm{P}(\mathbb{X} \,|\, \boldsymbol{w})$$

ML Estimation

- Log-likelihood:

$$\begin{aligned}\log \mathrm{P}(\mathbb{X}|\boldsymbol{w}) &= \log \textstyle\prod_{i=1}^{N} \mathrm{P}\left(\boldsymbol{x}^{(i)}, y^{(i)} \,|\, \boldsymbol{w}\right) \\ &= \log \textstyle\prod_i \mathrm{P}\left(y^{(i)} \,|\, \boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \mathrm{P}\left(\boldsymbol{x}^{(i)} \,|\, \boldsymbol{w}\right) \\ &\propto \log \textstyle\prod_i \sigma(\boldsymbol{w}^\top \boldsymbol{x}^{(i)})^{y'^{(i)}} [1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}^{(i)})]^{(1-y'^{(i)})} \\ &= \textstyle\sum_i y'^{(i)} \boldsymbol{w}^\top \boldsymbol{x}^{(i)} - \log(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^{(i)}}) \text{ [Homework]}\end{aligned}$$

- Unlike in linear regression, we cannot solve $\boldsymbol{w}$ analytically in a closed form via

$$\nabla_{\boldsymbol{w}} \log \mathrm{P}(\mathbb{X}|\boldsymbol{w}) = \sum_{i=1}^{N} [y'^{(i)} - \sigma(\boldsymbol{w}^\top \boldsymbol{x}^{(i)})] \boldsymbol{x}^{(i)} = \mathbf{0}$$

- But since $\log \mathrm{P}(\mathbb{X}|\boldsymbol{w})$ is differentiable w.r.t. $\boldsymbol{w}$, we can solve $\boldsymbol{w}^*_{\mathrm{ML}}$ numerically using stochastic gradient descent (SGD)
  - It can be shown that $\log \mathrm{P}(\mathbb{X}|\boldsymbol{w})$ is concave in terms of $\boldsymbol{w}$ [1]
  - SGD finds global optimal

Outline

MAP Estimation

- So far, we solve $\boldsymbol{w}$ by ML estimation:

$$\boldsymbol{w}_{\text{ML}} = \arg\max_{\boldsymbol{w}} \text{P}(\mathbb{X} \,|\, \boldsymbol{w})$$

- In MAP estimation, we solve

$$\boldsymbol{w}_{\text{MAP}} = \arg\max_{\boldsymbol{w}} \text{P}(\boldsymbol{w} \,|\, \mathbb{X}) = \arg\max_{\boldsymbol{w}} \text{P}(\mathbb{X} \,|\, \boldsymbol{w}) \text{P}(\boldsymbol{w})$$

- $\text{P}(\boldsymbol{w})$ models our ***preference*** or ***prior knowledge*** about $\boldsymbol{w}$

MAP Estimation for Linear Regression

- MAP estimation in linear regression:

$$\boldsymbol{w}_{\text{MAP}} = \arg\max_{\boldsymbol{w}} \log[\mathrm{P}(\mathbb{X}|\boldsymbol{w})\mathrm{P}(\boldsymbol{w})]$$

- If we assume that $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\boldsymbol{I})$

$$\begin{aligned}\log[\mathrm{P}(\mathbb{X}|\boldsymbol{w})\mathrm{P}(\boldsymbol{w})] &= \log \mathrm{P}(\mathbb{X}|\boldsymbol{w}) + \log \mathrm{P}(\boldsymbol{w}) \propto -\textstyle\sum_i \left(y^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)}\right)^2 \\ &\quad + \log \sqrt{\frac{1}{(2\pi)^D \det(\beta^{-1}\boldsymbol{I})}} \exp\left[-\tfrac{1}{2}(\boldsymbol{w}-\mathbf{0})^\top (\beta^{-1}\boldsymbol{I})^{-1}(\boldsymbol{w}-\mathbf{0})\right] \\ &\propto -\textstyle\sum_i \left(y^{(i)} - \boldsymbol{w}^\top \boldsymbol{x}^{(i)}\right)^2 - \beta \boldsymbol{w}^\top \boldsymbol{w}\end{aligned}$$

- $\mathrm{P}(\boldsymbol{w})$ corresponds to the ***weight decay*** term in Ridge regression
- MAP estimation provides a way to design complicated yet interpretable regularization terms
  - E.g., we have LASSO by letting $\mathrm{P}(\boldsymbol{w}) \sim \text{Laplace}(0, b)$ [Proof]
  - We can also let $\mathrm{P}(\boldsymbol{w})$ be a mixture of Gaussians

Remarks on ML and MAP Estimation

Theorem (Consistency)

*The ML estimator Θ_{ML} is **consistent**, i.e., $\lim_{N\to\infty} \Theta_{ML} \xrightarrow{\Pr} \Theta^*$ as long as the "true" $\mathrm{P}(y|\boldsymbol{x};\Theta^*)$ lies within our model $\mathbb{F}$.*

Theorem (Cramér-Rao Lower Bound [2])

At a fixed (large) number N of examples, no consistent estimator of $\hat{\Theta}$ has a lower expected MSE (mean square error) than the ML estimator Θ_{ML}.

- That is, Θ_{ML} has a low sample complexity (or is statistic efficient)
- ML estimation is popular due to its consistency and efficiency
- When N is small that yields overfitting behavior, we can use MAP estimation to ***introduce bias*** and ***reduce variance***

Outline

Bayesian Estimation

- In ML/MAP estimation, we use the estimated $\hat{\Theta}$ as a constant to make prediction:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}; \hat{\Theta})$$

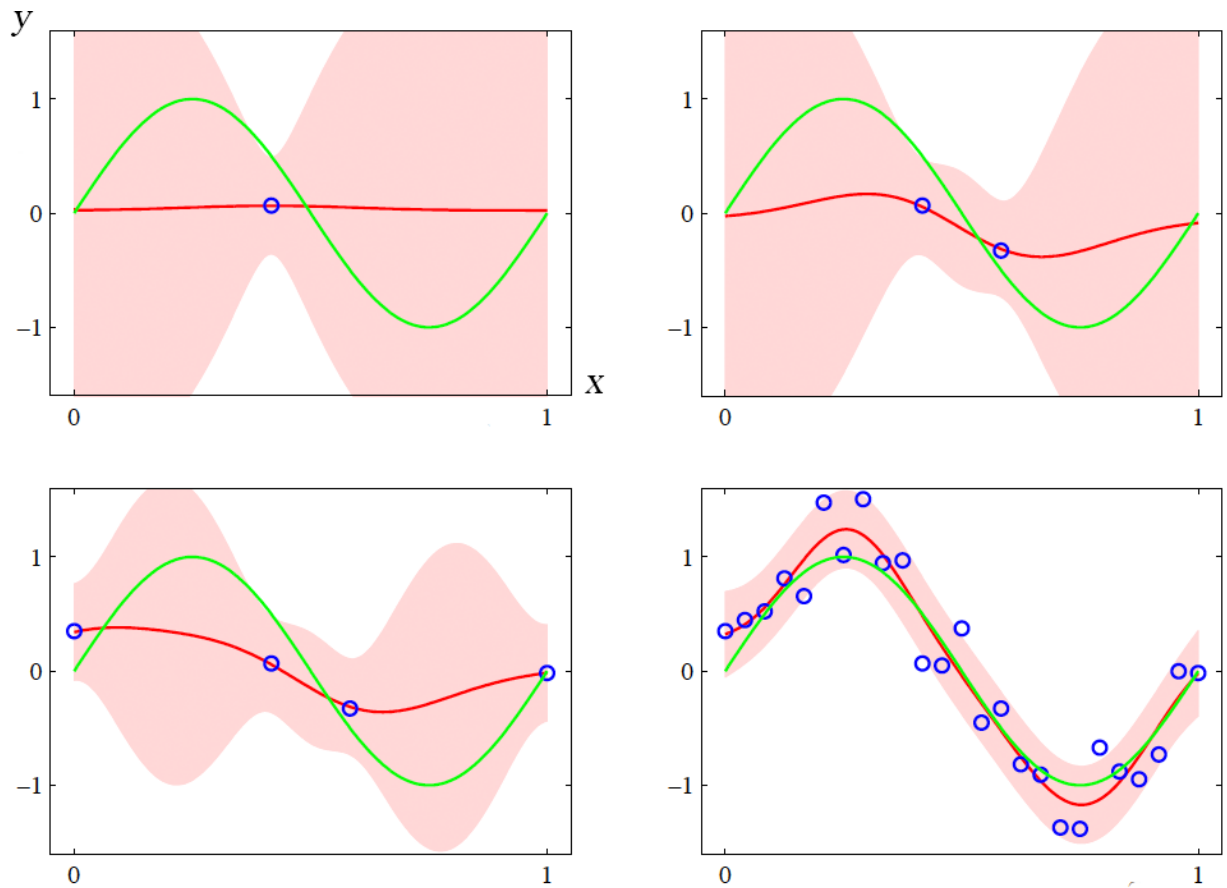
  - $\hat{\Theta} = \arg\max_{\Theta} \mathrm{P}(\Theta \,|\, \mathbb{X})$
- ***Bayesian inference*** threats Θ as a random variable when making prediction:

$$\mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathbb{X}) = \int_{\Theta} \mathrm{P}(\boldsymbol{y}, \Theta \,|\, \boldsymbol{x}, \mathbb{X}) d\Theta = \int \mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}, \Theta) \mathrm{P}(\Theta \,|\, \mathbb{X}) d\Theta$$

  - **Entire distribution** $\mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathbb{X})$ is calculated, so we get not only $\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathbb{X})$ but the uncertainty of each prediction
  - ***Bayesian estimation of Θ:*** each prediction considers ***all*** Θ's (weighted by their chances $\mathrm{P}(\Theta \,|\, \mathbb{X})$)

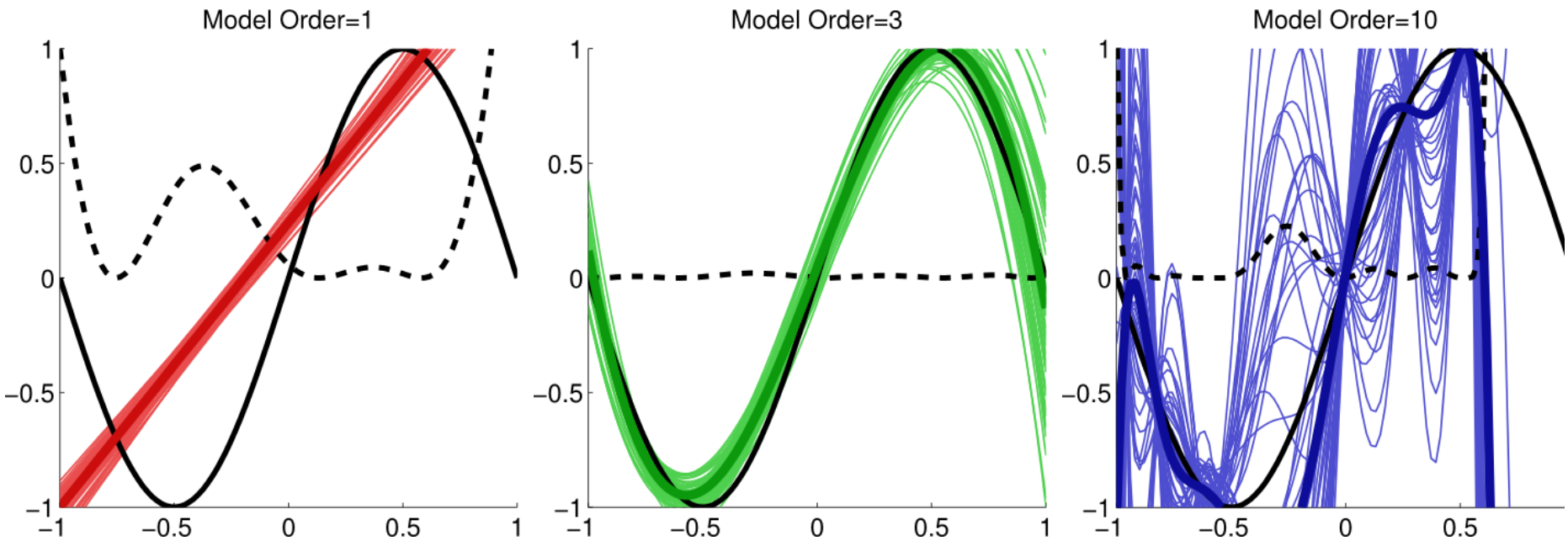
Example: 1D Regression

- Let $\mathrm{y} = f^*(\mathrm{x}) + \varepsilon$
  - Green line: $f^*(\cdot)$
  - Blue dots: noisy examples
- Red line: predictions by a Bayesian regressor (Gaussian Process)
- Shaded area: confidence intervals of predictions



Bayesian vs. ML Estimation

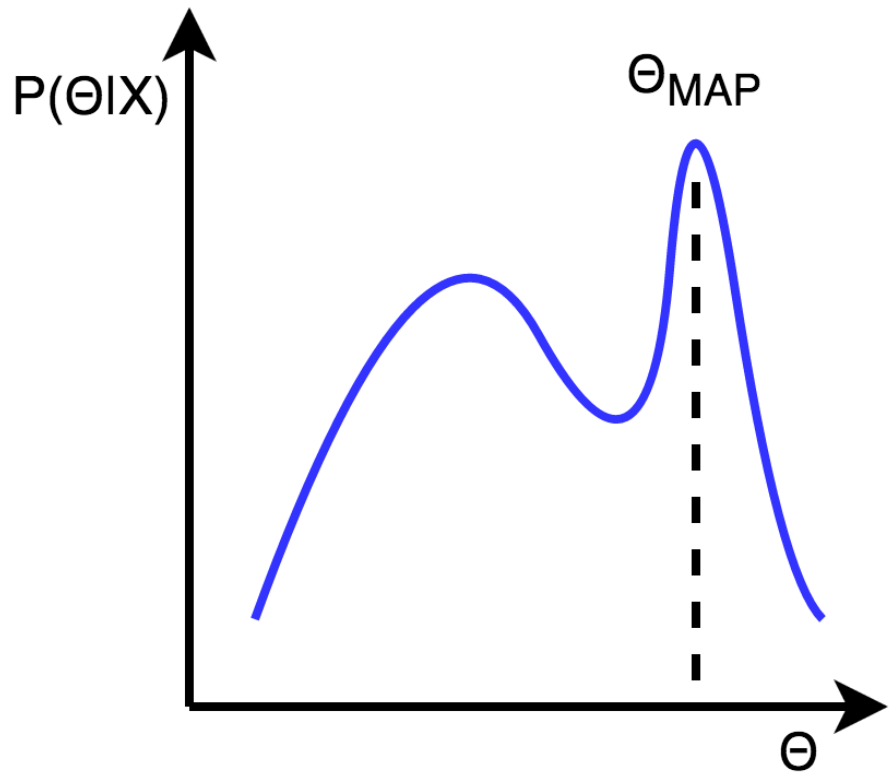
- Recall the bias-variance trade-off an ML-base polynomial regressor:



- Bayesian regressor usually generalizes better when the size N of training set is small
  - Avoids high variance $\mathrm{Var}_{\mathbb{X}}(\Theta_{\mathrm{ML}})$

Bayesian vs. MAP Estimation

- MAP gains some benefit of Bayesian approach by incorporating prior as $\text{bias}(\Theta_{\text{MAP}})$
  - Reduces $\text{Var}_{\mathbb{X}}(\Theta_{\text{MAP}})$ when training set is small
- However, does ***not*** work if Θ_{MAP} is unrepresentative of the majority Θ in $\int \text{P}(\boldsymbol{y}, \Theta | \boldsymbol{x}, \mathbb{X}) d\Theta = \int \text{P}(\boldsymbol{y} | \boldsymbol{x}, \Theta) \text{P}(\Theta | \mathbb{X}) d\Theta$
- E.g. when $\text{P}(\Theta | \mathbb{X})$ is a mixture of Gaussian



Evaluating $\mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \mathbb{X})$

$$\mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \mathbb{X}) = \int_{\Theta} \mathrm{P}(\boldsymbol{y}, \Theta \mid \boldsymbol{x}, \mathbb{X}) d\Theta = \int \mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \Theta) \mathrm{P}(\Theta \mid \mathbb{X}) d\Theta$$

- Integral computation make the evaluation challenging
  - The solution may not be tractable in many applications
- Fortunately, in the context of Bayesian linear regression, $\mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \mathbb{X})$ can have a simple, closed form [3]

Bayesian Linear Regression

- Assuming that $y = \boldsymbol{w}^\top \mathbf{x} + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, we have

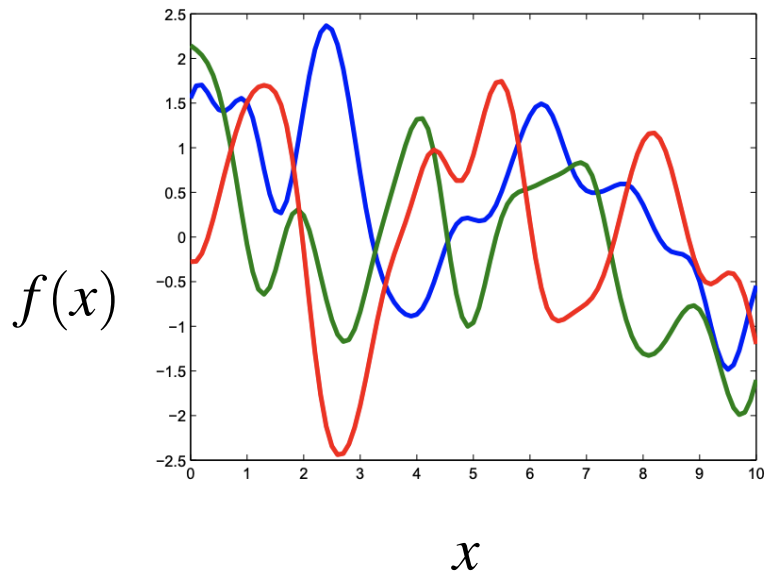
$$\begin{aligned} \mathrm{P}(y|\boldsymbol{x}, \mathbb{X}) &= \int_{\boldsymbol{w}} \mathrm{P}(y, \boldsymbol{w}|\boldsymbol{x}, \mathbb{X}) d\boldsymbol{w} = \int \mathrm{P}(y|\boldsymbol{x}, \boldsymbol{w}) \mathrm{P}(\boldsymbol{w}|\mathbb{X}) d\boldsymbol{w} \\ &= \frac{1}{\mathrm{P}(\mathbb{X})} \int \mathrm{P}(y|\boldsymbol{x}, \boldsymbol{w}) \mathrm{P}(\mathbb{X}|\boldsymbol{w}) \mathrm{P}(\boldsymbol{w}) d\boldsymbol{w} \\ &= \frac{\Pi_{i=1}^N \mathrm{P}(\boldsymbol{x}^{(i)})}{\mathrm{P}(\mathbb{X})} \int \mathrm{P}(y|\boldsymbol{x}, \boldsymbol{w}) \Pi_{i=1}^N \mathrm{P}(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{w}) \mathrm{P}(\boldsymbol{w}) d\boldsymbol{w} \end{aligned}$$

- $\mathrm{P}(y|\boldsymbol{x}, \boldsymbol{w}) \sim \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}, \sigma_\varepsilon)$ and $\mathrm{P}(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{w}) \sim \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}^{(i)}, \sigma_\varepsilon)$, $\forall i$
- If we assume that $\boldsymbol{w} \sim \mathcal{N}$, then $\mathrm{P}(y|\boldsymbol{x}, \boldsymbol{w}) \Pi_i \mathrm{P}(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{w}) \mathrm{P}(\boldsymbol{w})$ can be described by an $N+2$ dimensional Gaussian
  - $(y|\boldsymbol{x}, \boldsymbol{w})$, $(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{w})$, and $\boldsymbol{w}$ are (conditionally) independent with each other
- Its marginalization, $\mathrm{P}(y|\boldsymbol{x}, \mathbb{X}) = \int \mathrm{P}(y, \boldsymbol{w}|\boldsymbol{x}, \mathbb{X}) d\boldsymbol{w}$ is also a Gaussian
  - Gaussian distribution is closed under marginalization
- Why not model $(y|\boldsymbol{x}, \mathbb{X}) \sim \mathcal{N}$ in the first place?

Outline

Gaussian Process

- Assume a model $\mathbb{F}$ where the domain of each $f(\cdot) \in \mathbb{F}$ consists of only N inputs $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}$
- Let $y^{(i)} = f(\boldsymbol{x}^{(i)}) \in \mathbb{R}$, $\forall i$, we can compactly represent $f(\cdot)$ as a vector $\boldsymbol{y} = [y^{(1)}, \cdots, y^{(N)}]^\top$
- We can specify the probability of $f(\cdot)$ by assuming a distribution over $\boldsymbol{y}$, e.g., $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$
- A ***stochastic process*** is a random distribution over functions in $\mathbb{F}$
  - Alternatively, it can be a set of random random variables $\{\boldsymbol{y}^{(i)} \equiv f(\boldsymbol{x}^{(i)})\}_i$ indexed by time or space i



Gaussian Process (GP)

- A ***Gaussian process*** is a stochastic process of which the distribution is defined by a mean function $m(\cdot)$ and covariance/***kernel*** function $k(\cdot,\cdot)$:

$$\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{m} = \begin{bmatrix} m(\boldsymbol{x}^{(1)}) \\ \vdots \\ m(\boldsymbol{x}^{(N)}) \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(N)}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}^{(N)},\boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)},\boldsymbol{x}^{(N)}) \end{bmatrix}$$

- Intuition? If $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ are positively (or negatively) correlated, then $y^{(i)}$ and $y^{(j)}$ should be positively (or negatively) correlated too
- Common choices of mean and kernel functions:
  - $m(\cdot) = 0$
  - $k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) = \exp(-\frac{\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\|^2}{2\tau^2})$ for some fixed $\tau \in \mathbb{R} - \{0\}$
- The kernel matrix $\boldsymbol{K}$ is usually made positive definite (when $\boldsymbol{x}^{(i)} \neq \boldsymbol{x}^{(j)}, \forall i,j$) so it is invertible

Bayesian Regression

- Given N examples $\mathbb{X} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, how to predict the labels of M unlabeled instances $\mathbb{X}' = \{\boldsymbol{x}'^{(i)}\}_{i=1}^{M}$?
- Gaussian process:

$$\begin{bmatrix} \boldsymbol{y}_N \\ \boldsymbol{y}_M \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}_N \\ \boldsymbol{m}_M \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{N,N} & \boldsymbol{K}_{N,M} \\ \boldsymbol{K}_{M,N} & \boldsymbol{K}_{M,M} \end{bmatrix} \right),$$

  - $\boldsymbol{m}_N = \boldsymbol{m}_M = \boldsymbol{0}$ or $\bar{y}_N \boldsymbol{1}$, where $\bar{y}_N = \frac{1}{N}\Sigma_{i=1}^{N} y^{(i)}$
  - $\boldsymbol{y}_M$ is unknown
- Bayesian inference:

$$\mathrm{P}(\boldsymbol{y}_M \,|\, \mathbb{X}', \mathbb{X}) = \mathcal{N}(\boldsymbol{K}_{M,N}\boldsymbol{K}_{N,N}^{-1}\boldsymbol{y}_N \,, \boldsymbol{K}_{M,M} - \boldsymbol{K}_{M,N}\boldsymbol{K}_{N,N}^{-1}\boldsymbol{K}_{N,M})$$

  - Gaussian distribution is closed under conditioning
- There is ***no*** explicit training phase
- Predictions: $\hat{\boldsymbol{y}}_M = \boldsymbol{K}_{M,N}\boldsymbol{K}_{N,N}^{-1}\boldsymbol{y}_N$ (with uncertainty)

Noisy Data

- What if the examples $\mathbb{X} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ contain noise, i.e., $\mathrm{y} = f^*(\mathbf{x}) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$?
- From the i.i.d. noise assumption where $\varepsilon^{(i)}$ and $\varepsilon^{(j)}$ are independent, $\forall i, j$, we have

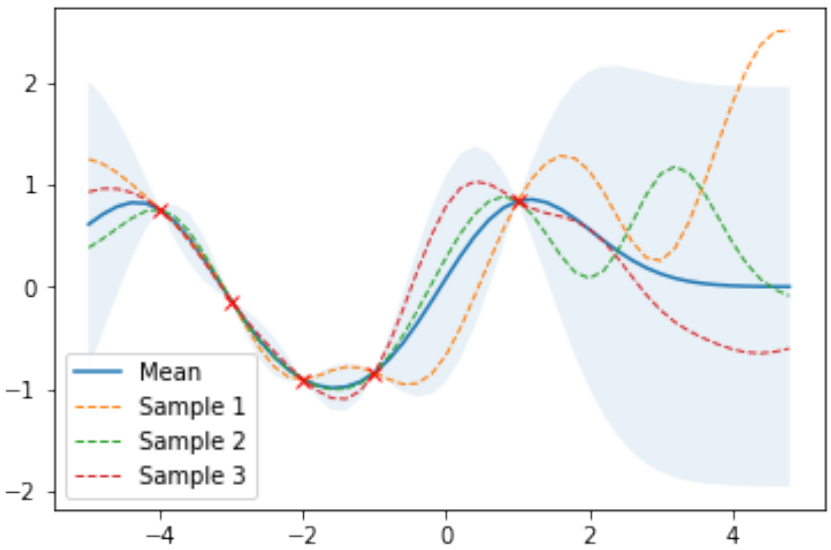
$$\begin{bmatrix} \boldsymbol{y}_N \\ \boldsymbol{y}_M \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}_N \\ \boldsymbol{m}_M \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{N,N} + \sigma_\varepsilon^2 \boldsymbol{I}_N & \boldsymbol{K}_{N,M} \\ \boldsymbol{K}_{M,N} & \boldsymbol{K}_{M,M} + \sigma_\varepsilon^2 \boldsymbol{I}_M \end{bmatrix} \right.$$

- Bayesian inference:

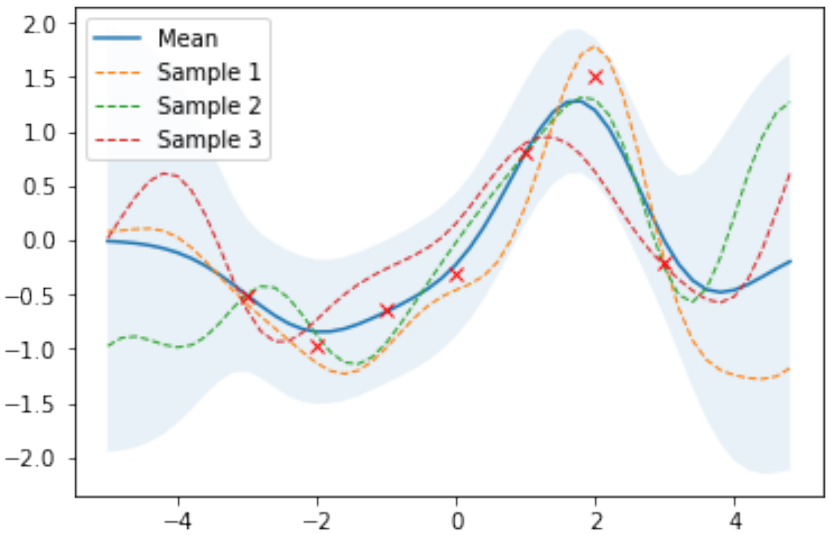
$$\begin{aligned} \mathrm{P}(\boldsymbol{y}_M \mid \mathbb{X}', \mathbb{X}) \quad &= \mathcal{N}(\boldsymbol{K}_{M,N}(\boldsymbol{K}_{N,N} + \sigma_\varepsilon^2 \boldsymbol{I}_N)^{-1} \boldsymbol{y}_N, \\ &\qquad \boldsymbol{K}_{M,M} + \sigma_\varepsilon^2 \boldsymbol{I}_M - \boldsymbol{K}_{M,N}(\boldsymbol{K}_{N,N} + \sigma_\varepsilon^2 \boldsymbol{I}_N)^{-1} \boldsymbol{K}_{N,M}) \end{aligned}$$

- Predictions: $\hat{\boldsymbol{y}}_M = \boldsymbol{K}_{M,N}(\boldsymbol{K}_{N,N} + \sigma_\varepsilon^2 \boldsymbol{I}_N)^{-1} \boldsymbol{y}_N$ (with uncertainty)

Predictions Given Clean and Noisy Data



Clean Data



Noisy Data

Other Choices of Kernels

- Radial basis function (RBF) or exponentiated quadratic kernel:

$$k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) = \exp(-\frac{\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\|^2}{2\tau^2})$$

  for some fixed $\tau \in \mathbb{R} - \{0\}$

- Periodic kernel:

$$k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) = \exp(-\frac{2}{\tau^2}\sin^2(\frac{\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\|}{p}\pi))$$

  for some fixed $\tau, p \in \mathbb{R} - \{0\}$
  - Suitable for sequential data with periodic patterns

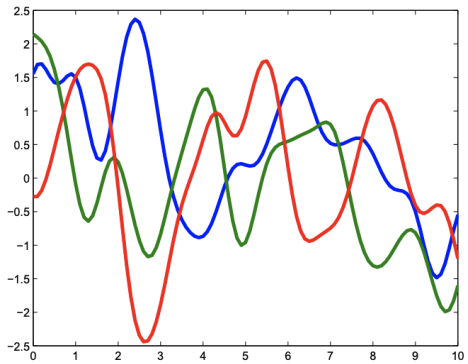
- Combined kernel:

$$k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) = k^{(1)}(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) \cdot k^{(2)}(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) \cdots$$

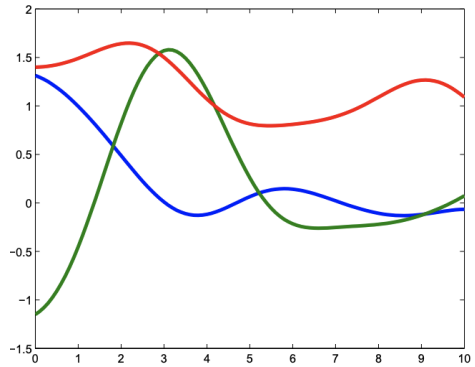
  - Has a high value only if all source covariances have a high value (AND operation)

Hyperparameter Tuning

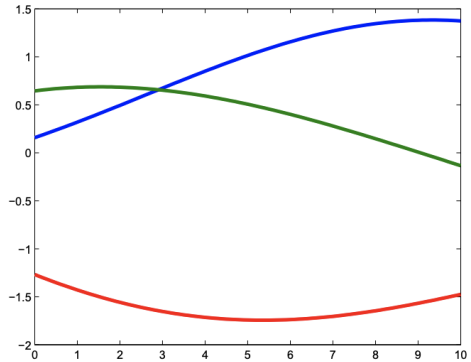
- The τ in the RBF kernel $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \exp(-\frac{\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|^2}{2\tau^2})$ controls the "smoothness" of the prediction functions



$\tau = 0.5$



$\tau = 2$



$\tau = 10$

- How to decide the best τ for a given $\mathbb{X}$?
  - More generally, how to decide the hyperparameters of chosen kernels?
- We can solve τ using the ML estimation we already familiar with:

$$\begin{aligned} \tau_{\text{ML}} &= \arg\min_\tau -\log \mathrm{P}(\mathbb{X} \,|\, \tau) = \arg\min_\tau -\log \mathrm{P}(\boldsymbol{y}_N \,|\, \boldsymbol{X}_N, \tau) \\ &= \arg\min_\tau (\boldsymbol{y}_N - \boldsymbol{m}_N)^\top \boldsymbol{K}_{N,N}^{-1} (\boldsymbol{y}_N - \boldsymbol{m}_N) + \log\det(\boldsymbol{K}_{N,N}) \end{aligned}$$

  - Derivable w.r.t. τ, so can be solved using a gradient-based approach

Parametric vs. Non-Parametric Models

- Probabilistic linear regression and logistic regression are special cases of ***parametric models***, whose #parameters is fixed with respect to #data seen
  - $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x}$ or $\hat{y} = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x})$
  - ***Model complexity grows with data dimension*** D
- Gaussian process, on the other hand, is a ***non-parametric model***
  - $\hat{\boldsymbol{y}}_M = \boldsymbol{K}_{M,N} \boldsymbol{K}_{N,N}^{-1} \boldsymbol{y}_N$, where each predicted label $\hat{y}$ is a linear combination of the labels in training set
  - ***Model complexity grows with*** N

Remarks

- Bayesian estimation:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathrm{P}(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathbb{X}) = \arg\max_{\boldsymbol{y}} \int \mathrm{P}(\boldsymbol{y}, \Theta \,|\, \boldsymbol{x}, \mathbb{X}) d\Theta$$

- Usually generalizes better given a small training set
- Unfortunately, solution may not be tractable in many applications
- Even tractable, incurs high computation cost
  - In GP, each batch of predictions $\hat{\boldsymbol{y}}_M = \boldsymbol{K}_{M,N} \boldsymbol{K}_{N,N}^{-1} \boldsymbol{y}_N$ may take $O(N^3)$ time
  - Not suitable for large-scale learning tasks

Reference I

[1] Deepak Roy Chittajallu.
Why is the error function minimized in logistic regression convex?
http://mathgotchas.blogspot.tw/2011/10/why-is-error-function-minimized-in.html, 2011.

[2] Harald Cramér.
Mathematical Methods of Statistics.
Princeton university press, 1946.

[3] Carl Edward Rasmussen.
Gaussian processes in machine learning.
In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.