

Introduction to Machine Learning - CS725
Instructor: Prof. Ganesh Ramakrishnan
Lecture 15 - Non Parametric Regression, RKHS

Non Parametric Regression

Basis function expansion and the Kernel trick: Additional Discussion 1

Consider regression function $f(\mathbf{x}) = \sum_{j=1}^p w_j \phi_j(\mathbf{x})$ with weight vector

\mathbf{w} estimated as $\mathbf{w}_{Pen} = \underset{\mathbf{w}}{\operatorname{argmin}} \mathcal{L}(\phi, \mathbf{w}, \mathbf{y}) + \lambda \Omega(\mathbf{w})$

It can be shown that for $p \in [0, \infty)$, under certain conditions on K , the following can be equivalent representations:

$$f(\mathbf{x}) = \sum_{j=1}^p w_j \phi_j(\mathbf{x}) \text{ and } f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

Kernel
expansion

¹Section 5.8.1 of Tibshi.

Basis fn expansion

Recall: The Representer Theorem & Reproducing Kernel Hilbert Space (RKHS)

- ① If $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$ and $K(\mathbf{x}', \mathbf{x}) = \phi^T(\mathbf{x})\phi(\mathbf{x}')$ then the solution $\mathbf{w}^* \in \mathfrak{R}^n$ to the following problem

$$(\mathbf{w}^*, b^*) = \arg \min_{\mathbf{w}, b} \sum_{i=1}^m \mathbf{E} \left(f(\mathbf{x}^{(i)}), y^{(i)} \right) + \Omega(\|\mathbf{w}\|_2)$$

can be always written as $\phi^T(\mathbf{x})\mathbf{w}^* + b = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}^{(i)})$, provided $\Omega(\|\mathbf{w}\|_2)$ is a monotonically increasing function of $\|\mathbf{w}\|_2$. \mathfrak{R}^n is the Hilbert space and $K(., \mathbf{x}) : \mathcal{X} \rightarrow \mathfrak{R}$ is the

Reproducing (RKHS) Kernel (let us just appreciate RKHS)

The Reproducing Kernel Hilbert Space (RKHS)

WE WILL TRY AND ONLY MOTIVATE RKHS WITHOUT GETTING CAUGHT UP IN DEFINITION OF HILBERT SPACE

Consider the set of functions $\mathcal{K} = \{K(., \mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ and let \mathcal{H} be the set of all functions that are **finite** linear combinations of functions in \mathcal{K} . That is, any function $h \in \mathcal{H}$ can be written as

$$\mathbf{h}(.) = \sum_{t=1}^T \alpha_t K(., \mathbf{x}_t) \text{ for some } T \text{ and } \mathbf{x}_t \in \mathcal{X}, \alpha_t \in \mathbb{R}. \text{ One can}$$

easily verify that \mathcal{H} is a vector space² with an inner product.

easy!

²Try it yourself. Prove that \mathcal{H} is closed under vector addition and (real) scalar multiplication.

Inner Product over RKHS \mathcal{H}

For any $g(\cdot) = \sum_{s=1}^S \beta_s K(\cdot, \underline{\mathbf{x}}'_s) \in \mathcal{H}$ and $h(\cdot) = \sum_{t=1}^T \alpha_t K(\cdot, \underline{\mathbf{x}}_t) \in \mathcal{H}$,

define the inner product³ (some similarity between $g(\cdot)$ & $h(\cdot)$)

$$\langle g(\cdot), h(\cdot) \rangle = \sum_{s=1}^S \sum_{t=1}^T \beta_s \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t)$$

$$g(\cdot) = [\dots \beta_1 \dots \beta_2 \dots \beta_3 \dots]$$

$$h(\cdot) = [\dots \alpha_1 \dots \alpha_2 \dots \dots]$$

³Again, you can verify that $\langle f, g \rangle$ is indeed an inner product following properties such as

Inner Product over RKHS \mathcal{H}

For any $g(\cdot) = \sum_{s=1}^S \beta_s K(\cdot, \mathbf{x}'_s) \in \mathcal{H}$ and $h(\cdot) = \sum_{t=1}^T \alpha_t K(\cdot, \mathbf{x}_t) \in \mathcal{H}$,
define the inner product³

$$\langle h, g \rangle = \sum_{s=1}^S \beta_s \sum_{t=1}^T \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) \quad (1)$$

Further simplifying (1),

$$\langle h, g \rangle = \sum_{s=1}^S \beta_s \sum_{t=1}^T \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) = \sum_{s=1}^S \beta_s \underbrace{\sum_{t=1}^T \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t)}_{h(\mathbf{x}'_s)} = \sum_{s=1}^S \beta_s \underbrace{h(\mathbf{x}'_s)}_{\sum_t \alpha_t g(\mathbf{x}_t)}$$

³Again, you can verify that $\langle f, g \rangle$ is indeed an inner product following properties such as

Inner Product over RKHS \mathcal{H}

$$\langle h, g \rangle = \sum_{s=1}^S \beta_s \sum_{t=1}^T \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) = \sum_{s=1}^S \beta_s h(\mathbf{x}_s)$$

One immediately observes that in the special case that $g(\cdot) = K(\cdot, \mathbf{x})$,

some single
specific pt x

$$g(\cdot) = \sum \beta_s K(\cdot, x_s)$$

where

$$\beta_s = 1 \text{ if } x_s' = x$$
$$\beta_s = 0 \text{ o/w}$$

$$\langle h, g \rangle = h(x) = \sum_t \alpha_t K(x, x_t)$$

Inner Product over RKHS \mathcal{H}

$$\langle h, g \rangle = \sum_{s=1}^S \beta_s \sum_{t=1}^T \alpha_t K(\mathbf{x}'_s, \mathbf{x}_t) = \sum_{s=1}^S \beta_s h(\mathbf{x}_s)$$

One immediately observes that in the special case that $g() = K(., \mathbf{x})$,

$$\langle h, K(., \mathbf{x}) \rangle = h(\mathbf{x}) \quad (3)$$

Orthogonal Decomposition

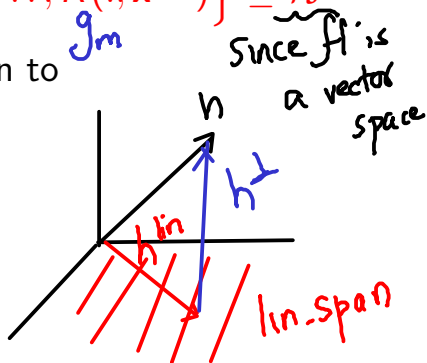
say training pt

Since $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}\} \subseteq \mathcal{X}$ and $\mathcal{K} = \{K(\cdot, \mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ with \mathcal{H} being the set of all finite linear combinations of functions in \mathcal{K} ,

$$\text{lin_span} \left\{ \underbrace{K(\cdot, \mathbf{x}^{(1)})}_{g_1}, \underbrace{K(\cdot, \mathbf{x}^{(2)})}_{g_2}, \dots, \underbrace{K(\cdot, \mathbf{x}^{(m)})}_{g_m} \right\} \subseteq \mathcal{H}$$

Thus, we can use orthogonal projection to

$$h = h^{\text{lin}} + h^{\perp}$$



Orthogonal Decomposition

Since $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}\} \subseteq \mathcal{X}$ and $\mathcal{K} = \{K(., \mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ with \mathcal{H} being the set of all finite linear combinations of functions in \mathcal{K} ,

$$\text{lin_span} \{K(., \mathbf{x}^{(1)}), K(., \mathbf{x}^{(2)}), \dots, K(., \mathbf{x}^{(m)})\} \subseteq \mathcal{H}$$

Thus, we can use orthogonal projection to decompose any $h \in \mathcal{H}$ into a sum of two functions, one lying in $\text{lin_span} \{...\}$, and the other lying in the **orthogonal complement**:

$$h = h^{\parallel} + h^{\perp} = \sum_{i=1}^m \alpha_i K(., \mathbf{x}^{(i)}) + h^{\perp} \quad (4)$$

where $\langle K(., \mathbf{x}^{(i)}), h^{\perp} \rangle = 0$, for each $i = [1..m]$.

For a specific training point $\mathbf{x}^{(j)}$, substituting from (4) into (3) for any $h \in \mathcal{H}$, using the fact that $\langle K(\cdot, \mathbf{x}^{(i)}), h^\perp \rangle = 0$

$$h(\mathbf{x}^{(j)}) = \left\langle \sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}^{(i)}) + h^\perp, K(\cdot, \mathbf{x}^{(j)}) \right\rangle$$
$$= \left\langle \sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}^{(i)}), K(\cdot, \mathbf{x}^{(j)}) \right\rangle$$

I expect h^\perp to be perpendicular (ie zero dot prod) to $K(\cdot, \mathbf{x}^{(i)})$

For a specific training point $\mathbf{x}^{(j)}$, substituting from (4) into (3) for any $h \in \mathcal{H}$, using the fact that $\langle K(., \mathbf{x}^{(i)}), h^\perp \rangle = 0$

$$\begin{aligned} h(\mathbf{x}^{(j)}) &= \left\langle \sum_{i=1}^m \alpha_i K(., \mathbf{x}^{(i)}) + h^\perp, K(., \mathbf{x}^{(j)}) \right\rangle \\ &= \sum_{i=1}^m \alpha_i \underbrace{\langle K(., \mathbf{x}^{(i)}), K(., \mathbf{x}^{(j)}) \rangle}_{g(.)} = \sum_{i=1}^m \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \end{aligned} \quad (5)$$

which we observe is independent of h^\perp .

Given $K(\cdot, \cdot)$ if space of linear regression functions is restricted to linear combinations of $K(\cdot, \cdot)$ with second argument fixated on any subset of pts from input space X then the linear regression is evaluated at a training point x_i is of the form of linear combinations of $K(x^{(i)}, x^{(j)})$ for $j=1 \dots m$

THEREFORE THE NAME REPRODUCING KERNEL (HILBERT SPACE)

Basis function expansion and the Kernel trick:

Additional Discussion 2

Consider regression function $f(\mathbf{x}) = \sum_{j=1}^p w_j \phi_j(\mathbf{x})$ with weight vector

\mathbf{w} estimated as $\mathbf{w}_{Pen} = \underset{\mathbf{w}}{\operatorname{argmin}} \mathcal{L}(\phi, \mathbf{w}, \mathbf{y}) + \lambda \Omega(\mathbf{w})$

It can be shown that for $p \in [0, \infty)$, under certain conditions on K , the following can be equivalent representations:

$$f(\mathbf{x}) = \sum_{j=1}^p w_j \phi_j(\mathbf{x}) \text{ and }^4 f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

⁴Section 5.8.1 of Tibshi.

Basis function expansion & Kernel: Additional Discussion 2

- We could also begin with (Eg: Nadaraya-Watson kernel regression)

$$f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^m y_i k_n(\|\mathbf{x} - \mathbf{x}_i\|)}{\sum_{i=1}^m k_n(\|\mathbf{x} - \mathbf{x}_i\|)}$$

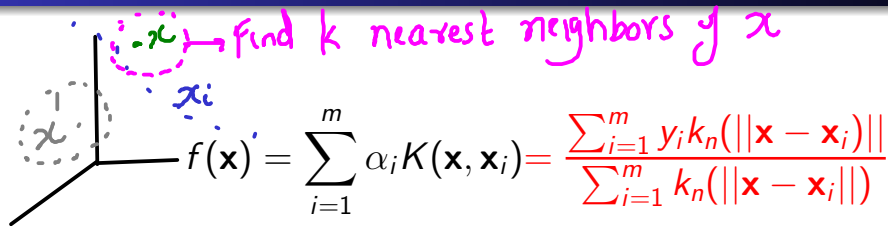
$k_n: \mathbb{R} \rightarrow \mathbb{R}^+$

A non-parametric kernel k_n is a non-negative real-valued integrable function satisfying the following two requirements:

$$\int_{-\infty}^{+\infty} k_n(u) du = 1 \text{ and } k_n(-u) = k_n(u) \text{ for all values of } u$$

Behaves as density

Basis function expansion & Kernel: Additional Discussion 2


$$f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^m y_i k_n(||\mathbf{x} - \mathbf{x}_i||)}{\sum_{i=1}^m k_n(||\mathbf{x} - \mathbf{x}_i||)}$$

- E.g.: $k_n(x_i - \textcircled{x}) = I(||x_i - x|| \leq ||x_{(k)} - x||)$ where $x_{(k)}$ is the training observation ranked k^{th} in distance from x and $I(S)$ is the indicator of the set S
- This is precisely the Nearest Neighbor Regression model

Another possibility $k_n(\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{\theta^2}{2\sigma^2})$

Basis function expansion & Kernel: Additional Discussion 2

$$f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i) = \frac{\sum_{i=1}^m y_i k_n(\|\mathbf{x} - \mathbf{x}_i\|)}{\sum_{i=1}^m k_n(\|\mathbf{x} - \mathbf{x}_i\|)}$$

- Kernel regression and density models are other examples of such *local regression* methods⁵
- The broader class - **Non-Parametric Regression:**
 $y = g(\mathbf{x}) + \epsilon$ where functional form of $g(\mathbf{x})$ is not fixed

⁵Section 2.8.2 of Tibshi

May not have fixed form
w/ $\phi(\mathbf{x})$

Non-parametric Kernel weighted (Local Linear) Regression: Tut 5, Prob 3

Given $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_n, y_n)\}$, predict $f(\mathbf{x}') = (\mathbf{w}'^\top \phi(\mathbf{x}') + b)$ for each test (or query point) \mathbf{x}' as:

$$(\mathbf{w}', b') = \operatorname{argmin}_{\mathbf{w}, b} \sum_{i=1}^n \underbrace{K(\mathbf{x}', \mathbf{x}_i)}_{w_i} \underbrace{(y_i - (\mathbf{w}^\top \phi(\mathbf{x}_i) + b))}_{\text{function of } \mathbf{x}'}^2$$

- 1 If there is a closed form expression for (\mathbf{w}', b') and therefore for $f(\mathbf{x}')$ in terms of the known quantities, derive it.
- 2 How does this model compare with linear regression and k-nearest neighbor regression? What are the relative advantages and disadvantages of this model?

Non-parametric Kernel weighted (Local Linear) Regression: Tut 5, Prob 3

3. In the one dimensional case (that is when $\phi(x) \in \mathbb{R}$), graphically try and interpret what this regression model would look like, say when $K(.,.)$ is the linear kernel⁶.

⁶Hint: What would the regression function look like at each training data point?

Answer to Question 1

The weighing factor $r_i^{x'}$ of each training data point (\mathbf{x}_i, y_i) is now also a function of the query or test data point $(\mathbf{x}', ?)$, so that we write it as $r_i^{x'} = K(\mathbf{x}', \mathbf{x}_i)$ for $i = 1, \dots, m$. Let $r_{m+1}^{x'} = 1$ and let R be an $(m+1) \times (m+1)$ diagonal matrix of $r_1^{x'}, r_2^{x'}, \dots, r_{m+1}^{x'}$.

weight matrix $\leftarrow R = \begin{bmatrix} r_1^{x'} & 0 & \dots & 0 \\ 0 & r_2^{x'} & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ 0 & 0 & 0 & \dots & r_{m+1}^{x'} \end{bmatrix}$ \rightarrow Diagonal since no correlation between pts explored

Further, let

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \dots & \phi_p(x_1) & 1 \\ \dots & \dots & \dots & 1 \\ \dots & \dots & \dots & 1 \end{bmatrix}$$

Answer to Question 1 (contd.)

$$\hat{\mathbf{w}}^l = \begin{bmatrix} w_1 \\ \dots \\ w_p \\ b \end{bmatrix}$$

and

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \dots \\ y_m \end{bmatrix}$$

The sum-square error function then becomes....

Answer to Question 1 (contd.)

The sum-square error function then becomes

$$\frac{1}{2} \sum_{i=1}^m r_i (y_i - (\hat{\mathbf{w}}^T \phi(x_i) + b))^2 = \frac{1}{2} \|\sqrt{R}\mathbf{y} - \sqrt{R}\Phi\hat{\mathbf{w}}\|_2^2$$

where \sqrt{R} is a diagonal matrix such that each diagonal element of \sqrt{R} is the square root of the corresponding element of R .

Answer to Question 1 (contd.)

The sum-square error function: NON-PARAMETRIC BECAUSE YOU HAVE TO REMEMBER EACH TRAINING DATA POINT FOR ANSWERING QUERY ON EACH NEW TEST POINT

$$\frac{1}{2} \sum_{i=1}^m r_i (y_i - (\hat{\mathbf{w}}^T \phi(x_i) + b))^2 = \frac{1}{2} \|\sqrt{R}\mathbf{y} - \sqrt{R}\Phi\hat{\mathbf{w}}\|_2^2$$

This convex function has a global minimum at $\hat{\mathbf{w}}_*^{x'}$ such that

$$\hat{\mathbf{w}}_*^{x'} = (\Phi^T R \Phi)^{-1} \Phi^T R \mathbf{y}$$

This is referred to as local linear regression (Section 6.1.1 of Tibshi).

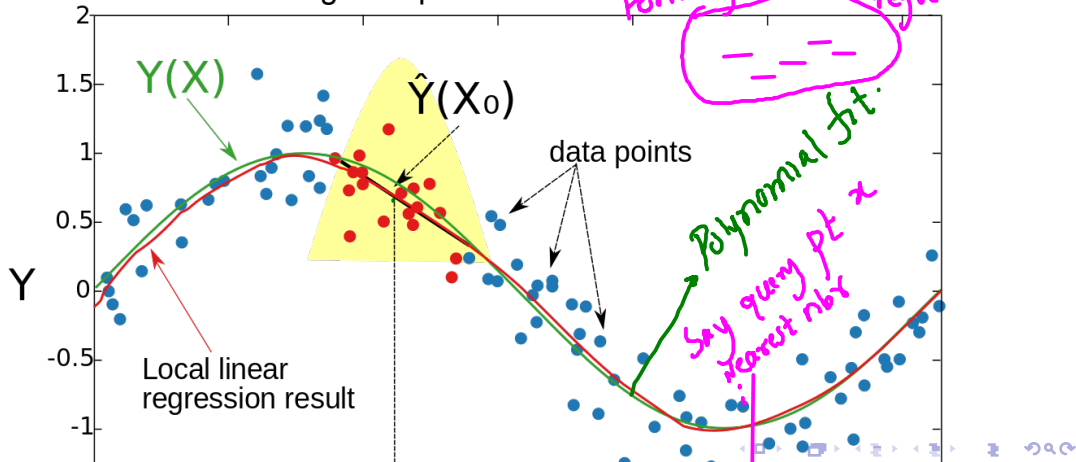
$$\hat{\mathbf{w}}_*^{x'} = (\sqrt{R^{x'}} \Phi)^T (\sqrt{R^{x'}} \Phi)^{-1} (\sqrt{R^{x'}} \Phi)^T (\sqrt{R^{x'}}) \mathbf{y}$$

Answer to Question 2

- 1 Local linear regression gives more importance (than linear regression) to points in \mathcal{D} that are closer/similar to \mathbf{x}' and less importance to points that are less similar.
- 2 Important if the regression curve is supposed to take different shapes in different parts of the space.
- 3 Local linear regression comes close to k-nearest neighbor. But unlike k-nearest neighbor, local linear regression gives you a smooth solution

Answer to Question 3

Every point on x axis is a potential query point. The local linear regression curve does not have any fixed (such as polynomial) form. It changes with the number of training data points



	Ridge Reg	Linear Reg.	Local Linear Reg	Nearest nbr Reg
--	-----------	-------------	------------------	-----------------

High Bias				
-----------	--	--	--	--

				Low Bias
--	--	--	--	----------

Low Variance				
--------------	--	--	--	--

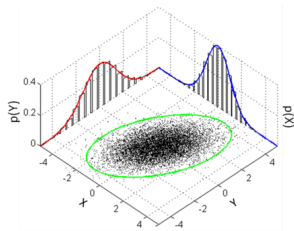
				High variance
--	--	--	--	---------------

Gaussian Process Regression

(Not for midsem)

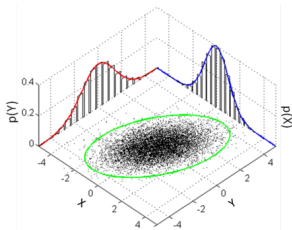
Gaussian Process: Definition and Example

- 1 For any set \mathcal{I} , a Gaussian Process (GP) on \mathcal{I} is a set of random variables $\{z_i \mid i \in \mathcal{I}\}$ such that, for any $n \in \mathcal{N}$, $i_1, \dots, i_n \in \mathcal{I}$, $\mathbf{z}_n = \{z_{i_1}, \dots, z_{i_n}\}$ is multivariate Gaussian:
- $$\mathcal{N}(\mathbf{z}_n; \mu_{\mathbf{z}_n}, \Sigma_{\mathbf{z}_n}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{\mathbf{z}_n}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{z}_n - \mu_{\mathbf{z}_n})^T \Sigma_{\mathbf{z}_n}^{-1} (\mathbf{z}_n - \mu_{\mathbf{z}_n})} \text{ when}$$
- $\Sigma_{\mathbf{z}_n} \in \mathbb{R}^{n \times n}$ is positive-definite and $\mu_{\mathbf{z}_n} \in \mathbb{R}^n$



- 2 Eg of Random Lines:.....(next slide)

Gaussian Process: Definition and Example



Uncountably infinite
index set.

$$Z = [0.5s, 0.6s, 8s, \dots]$$

- 1 Eg of Random Lines: $\mathcal{I} = \mathbb{R}$, $z_i = iS$ where $S \sim \mathcal{N}(0, 1)$.
Then, $\mu_{z_n} = [i_1, \dots, i_n]^T \times 0 = \mathbf{0} \in \mathbb{R}^n$ and
 $\Sigma_{z_n} = \text{diag}(i_1, \dots, i_n)$
- 2 In fact, it can be proved that for any choice of mean and covariance (functions), there exists a corresponding Gaussian Process

Gaussian Process: Kernels and Covariance Matrices

- 1 Both the covariance and Kernel are functions of two arguments
- 2 Both measure similarity between pairs of arguments
- 3 Both the covariance matrix Σ_{z_n} and the kernel (Gram) matrix K_{z_n} are to be symmetric, positive semi-definite matrices
These are also referred to often as **Positive Definite (!!!)** or **Autocovariance** kernels

Gaussian Process: Kernels and Covariance Matrices

Positive Definite (!!!) or **Autocovariance** kernels

More Examples:

- 1 **Linear:** $\mathcal{I} = \mathbb{R}$, $\mu_{i_p} = 0$, $cov(z_{i_p}, z_{i_q}) = \underline{K(z_{i_p}, z_{i_q})} = z_{i_p} z_{i_q}$
- 2 **Brownian:** $\mathcal{I} = \mathbb{R}$,
 $\mu_{i_p} = 0$, $cov(z_{i_p}, z_{i_q}) = K(z_{i_p}, z_{i_q}) = \min(z_{i_p}, z_{i_q})$
- 3 **Exponential:** $\mathcal{I} = \mathbb{R}$,
✓ $\mu_{i_p} = 0$, $cov(z_{i_p}, z_{i_q}) = \underline{K(z_{i_p}, z_{i_q})} = \exp(-\gamma(z_{i_p} - z_{i_q})^2)$
- 4 **Periodic:** $\mathcal{I} = \mathbb{R}$,
 $\mu_{i_p} = 0$, $cov(z_{i_p}, z_{i_q}) = K(z_{i_p}, z_{i_q}) = \exp(-\gamma \sin(\pi(z_{i_p} - z_{i_q}))^2)$

Bayesian Linear Regression: Recap

- 1 We began with Gaussian noise: $y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- 2 With prior on \mathbf{w} as $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$, we get $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$ where, $\Sigma_m^{-1} = \lambda I + \Phi^T \Phi / \sigma^2$ and $\mu_m = (\lambda \sigma^2 I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- 3 Tutorial 3 & 4: We showed that the solution $\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{w} | \mathcal{D})$ is the same as

$$\operatorname{argmin} \frac{1}{2} \|\Phi \mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2$$

Bayesian Linear Regression: Recap

- 1 We began with Gaussian noise: $y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- 2 With prior on \mathbf{w} as $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$, we get $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$ where, $\Sigma_m^{-1} = \lambda I + \Phi^T \Phi / \sigma^2$ and $\mu_m = (\lambda \sigma^2 I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- 3 Tutorial 3 & 4: We showed that the solution $\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{w} | \mathcal{D})$ is the same as that of *Regularized Ridge Regression*: $\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\phi \mathbf{w} - \mathbf{y}\|_2^2 + \lambda \sigma^2 \|\mathbf{w}\|_2^2$
- 4 Tutorial 5: We derived its

Bayesian Linear Regression: Recap

- 1 We began with Gaussian noise: $y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- 2 With prior on \mathbf{w} as $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$, we get $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$ where, $\Sigma_m^{-1} = \lambda I + \Phi^T \Phi / \sigma^2$ and $\mu_m = (\lambda \sigma^2 I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- 3 Tutorial 3 & 4: We showed that the solution $\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{w} | \mathcal{D})$ is the same as that of *Regularized Ridge Regression*: $\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\phi \mathbf{w} - \mathbf{y}\|_2^2 + \lambda \sigma^2 \|\mathbf{w}\|_2^2$
- 4 Tutorial 5: We derived its kernelized form for $K = \Phi \Phi^T$: $f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i)$, where $\alpha_i = ([K(\mathbf{x}_i, \mathbf{x}_j)] + \lambda I)^{-1} \mathbf{y}$;

Bayesian Linear Regression and Gaussian Process

- 1 We began with Gaussian noise: $y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- 2 With prior on \mathbf{w} as $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$, we get $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$ where, $\Sigma_m^{-1} = \lambda I + \Phi^T \Phi / \sigma^2$ and $\mu_m = (\lambda \sigma^2 I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- 3 **Viewing $\mathcal{I} = \{\phi(\cdot)\} = \Re^n$ and $z_x = \mathbf{w}^T \phi(\mathbf{x})$, we get z_x to be a (Linear) Gaussian Process on \Re^n .**
- 4 That is,

Bayesian Linear Regression and Gaussian Process

- 1 We began with Gaussian noise: $y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$
- 2 With prior on \mathbf{w} as $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$, we get $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$ where, $\Sigma_m^{-1} = \lambda I + \Phi^T \Phi / \sigma^2$ and $\mu_m = (\lambda \sigma^2 I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
- 3 **Viewing $\mathcal{I} = \{\phi(\cdot)\} = \mathbb{R}^n$ and $z_x = \mathbf{w}^T \phi(\mathbf{x})$, we get z_x to be a (Linear) Gaussian Process on \mathbb{R}^n .**
- 4 **That is, $[z_{x_1}, z_{x_2}, \dots, z_{x_m}]^T = \Phi \mathbf{w}$ is a multivariate Gaussian for every training set, with mean zero and covariance $\Phi \Phi^T$ (the kernel matrix)!!**

Gaussian Process Regression

- 1 We are given some training points $\mathcal{D}_{tr} = (\mathbf{x}_1, y_1) \dots (\mathbf{x}_m, y_m)$ and need to make predictions y'_1, \dots, y'_t on some test points $\mathcal{D}_{te} = \{\mathbf{x}'_1 \dots \mathbf{x}'_t\}$
- 2 We begin with a Gaussian process:

Gaussian Process Regression

- 1 We are given some training points $\mathcal{D}_{tr} = (\mathbf{x}_1, y_1) \dots (\mathbf{x}_m, y_m)$ and need to make predictions y'_1, \dots, y'_t on some test points $\mathcal{D}_{te} = \{\mathbf{x}'_1 \dots \mathbf{x}'_t\}$
- 2 We begin with a Gaussian process: $z_{\mathbf{x}} \sim GP(\mu, \underline{K})$ **defined at each point \mathbf{x}**
- 3 ...and Gaussian noise: $y = z_{\mathbf{x}} + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is independent of $z_{\mathbf{x}}$
- 4 We are interested in distribution of $\Pr([y'_1, \dots, y'_t] \mid [y_1, \dots, y_m])$ = $\mathcal{N}(\mu', \Sigma')$ where

$$\mu' = \mu_{te} + \underline{K}_{te,tr}(\underline{K}_{tr,tr} + \sigma^2 I)^{-1}(y_{tr} - \mu_{tr})$$
$$\Sigma' = (K_{te,te} + \sigma^2 T) - K_{tr,tr}(K_{te,te} + \sigma^2 I)^{-1}K_{tr,te}$$