Bayesian Linear Regression (Contd)

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

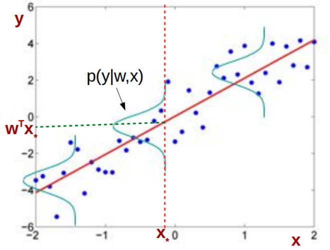
Jan 16, 2018

Linear Regression: A Probabilistic Setup

- Given: N training examples $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, features: $\boldsymbol{x}_n \in \mathbb{R}^D$, response $y_n \in \mathbb{R}$
- Assume a "noisy" linear model with regression weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_D] \in \mathbb{R}^D$

$$y_n = \boldsymbol{w}^\top \boldsymbol{x}_n + \epsilon_n$$

  where $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$, β: precision (inverse variance) of Gaussian (assumed known)
- Therefore $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$



- Note: Some books (e.g., PRML) use $\phi(\boldsymbol{x}_n)$ to denote the features where ϕ is some transformation of the original features $\boldsymbol{x}_n$ (we will only use this notation when talking about nonlinear regression)

The Likelihood Model

- Notation: $\mathbf{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_N]^\top$: $N \times D$ feature matrix, $\boldsymbol{y} = [y_1 \ldots y_N]^\top$: $N \times 1$ response vector
- Assuming i.i.d. observations, the likelihood model

$$
\begin{aligned}
p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta) = \prod_{n=1}^{N} p(y_n|\boldsymbol{w}, \boldsymbol{x}_n, \beta) &= \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) \\
&= \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2\right] \\
&= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left[-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2\right]
\end{aligned}
$$

- Can also write the likelihood $p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X})$ as an N-dim multivariate Gaussian

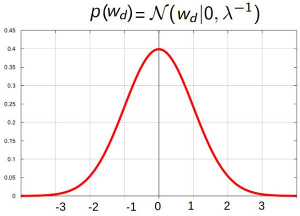
$$
p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left[-\frac{\beta}{2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{w})\right]
$$

- Note that NLL = sum of squared errors! Minimizing w.r.t. $\boldsymbol{w}$ will give MLE/least squares solution!

The Prior

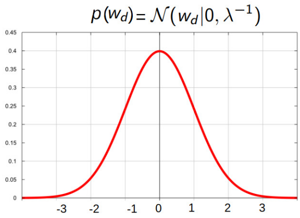
- Assume the entries in $\boldsymbol{w}$ are i.i.d. with zero mean Gaussian priors. Therefore

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} p(w_d) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda^{-1}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\mathbf{I}_D) = \left(\frac{\lambda}{2\pi}\right)^{\frac{D}{2}} \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}\right]$$



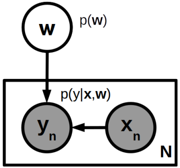
- This prior promotes the entries in $\boldsymbol{w}$ to be small (close to zero)
  - Also, the negative of log-prior is the same as an ℓ_2 regularizer on $\boldsymbol{w}$
- This prior is conjugate to the likelihood (Gaussian) which makes posterior inference easy

The Prior



- The role of the precision hyperparam λ in the prior is important
- Large values of λ would more aggressively encourage w_d to be close to zero
- Can think of λ as the regularization hyperparam for the weights
- Important: Can infer λ as well (will see later how to do this)
- Can even have different λ for each w_d, i.e., $p(\boldsymbol{w}|\{\lambda_d\}_{d=1}^D) = \prod_{d=1}^D \mathcal{N}(w_d|0, \lambda_d^{-1})$
  - Useful in sparse regression/classification models in which very few features are relevant which can be identified by inferring $\{\lambda_d\}_{d=1}^D$. More on this when we talk about sparse models.

Inference Tasks for Bayesian Linear Regression



(Hyperparameters λ,β not shown as they are fixed/known)

- Want to infer the posterior distribution over $\boldsymbol{w}$ (for now, assume β and λ to be known)

$$p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) = \frac{p(\boldsymbol{w}|\lambda)p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta)}{p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)}$$

- Want to infer the posterior predictive distribution

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \int p(y_*|\boldsymbol{w}, \boldsymbol{x}_*, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)d\boldsymbol{w}$$

- Since the likelihood model $p(y|\boldsymbol{w}, \boldsymbol{x}, \beta)$ and the prior $p(\boldsymbol{w}|\lambda)$ both are Gaussians, the above computations can be easily done

Brief Detour: Some Useful Properties of Multivariate Gaussians

Multivariate Gaussian Distribution

- The (multivariate) Gaussian with mean $\boldsymbol{\mu}$ and cov. matrix $\boldsymbol{\Sigma}$

$$
\begin{aligned}
\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \\
&= \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}\text{trace}\left[\boldsymbol{\Sigma}^{-1}\mathbf{S}\right]\right\} \qquad \text{where } \mathbf{S} = (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top
\end{aligned}
$$

- An alternate representation: The "information form"

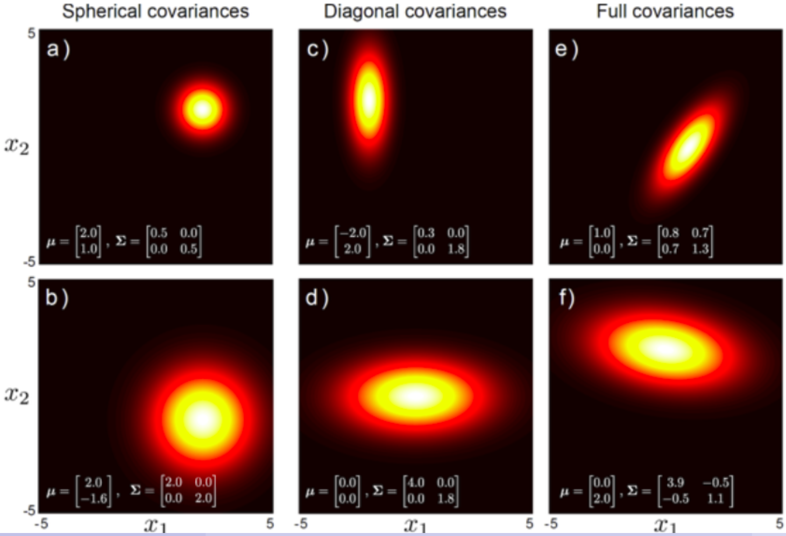
$$
\mathcal{N}_c(\boldsymbol{x}|\boldsymbol{\xi},\boldsymbol{\Lambda}) = (2\pi)^{-D/2}|\boldsymbol{\Lambda}|^{1/2}\exp\left\{-\frac{1}{2}\left(\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{x}+\boldsymbol{\xi}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}-2\boldsymbol{x}^\top\boldsymbol{\xi}\right)\right\}
$$

  where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ (precision matrix) and $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ are the "natural parameters"

- Note that there is a term quadratic in $\boldsymbol{x}$ (involves $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$) and linear in $\boldsymbol{x}$ (involves $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$)
- Information form can help recognize $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a Gaussian when doing algebraic manipulations

Multivariate Gaussian: The Covariance Matrix

The covariance matrix can be spherical, diagonal, or full



Marginals and Conditionals from Gaussian Joint Distribution

- Given $\boldsymbol{x}$ having multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. Suppose

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \qquad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{bmatrix}$$

- The marginal distribution of one block, say $\boldsymbol{x}_a$, is a Gaussian

$$p(\boldsymbol{x}_a) = \int p(\boldsymbol{x}_a, \boldsymbol{x}_b) d\boldsymbol{x}_b = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$$

- The conditional distribution of $\boldsymbol{x}_a$ given $\boldsymbol{x}_b$, is Gaussian, i.e., $p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$ where

$$\begin{aligned} \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \qquad (\text{“smaller” than } \Sigma_{aa}\text{; makes sense intuitively}) \\ \boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b}\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)\} \\ &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \\ &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \end{aligned}$$

- Both results are extremely useful especially when working with Gaussian joint distributions

Linear Gaussian Model

- Consider linear transformation of a Gaussian r.v. $\boldsymbol{z}$ with $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$, plus Gaussian noise

$$\boxed{\boldsymbol{x} = \mathbf{A}\boldsymbol{z} + \boldsymbol{b} + \epsilon} \qquad \text{where} \quad p(\epsilon) = \mathcal{N}(\epsilon|\mathbf{0}, \mathbf{L}^{-1})$$

- Easy to see that, conditioned on $\boldsymbol{z}$, $\boldsymbol{x}$ too has a Gaussian distribution

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathbf{A}\boldsymbol{z} + \boldsymbol{b}, \mathbf{L}^{-1})$$

- This is called a Linear Gaussian Model. Very commonly encountered in probabilistic modeling
- The following two distributions are of particular interest. Defining $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^\top \mathbf{L}\mathbf{A})^{-1}$, we have

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{z})} = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\Sigma}\left\{\mathbf{A}^\top \mathbf{L}(\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\right\}, \boldsymbol{\Sigma}) \qquad \text{(Gaussian posterior)}$$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \mathcal{N}(\boldsymbol{x}|\mathbf{A}\boldsymbol{\mu} + \boldsymbol{b}, \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top + \mathbf{L}^{-1}) \qquad \text{(Gaussian predictive/marginal)}$$

- Exercise: Prove the above two results (MLAPP Chap. 4 and PRML Chap. 2 contain the proof)
  - Can derive by first writing down the joint $p(\boldsymbol{x}, \boldsymbol{z})$, identifying its mean and covariance/precision matrix (using information form), and then apply conditioning and marginal formulas from previous slide

Bayesian Linear Regression: The Posterior

- The posterior over $\boldsymbol{w}$ (for now, assume hyperparams β and λ to be known)

$$p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) = \frac{p(\boldsymbol{w}|\lambda)p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta)}{p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)} \propto p(\boldsymbol{w}|\lambda)p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta)$$

- Computing $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)$

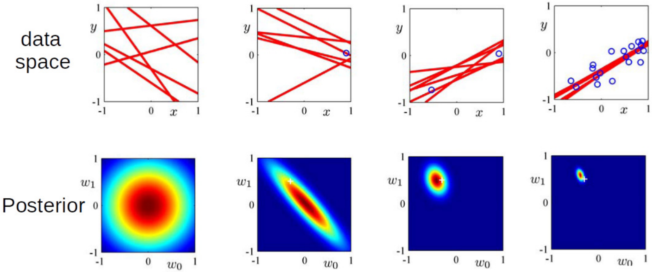
$$p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) \propto \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D) \times \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$$

- Using the "completing the squares" trick (or directly using Gaussian conditioning formula)

$$\begin{aligned}
p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) &= \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \\
\text{where} \quad \boldsymbol{\Sigma}_N &= (\beta \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda \mathbf{I}_D)^{-1} = (\beta \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \quad \text{(posterior's covariance matrix)} \\
\boldsymbol{\mu}_N &= \boldsymbol{\Sigma}_N \left[\beta \sum_{n=1}^{N} y_n \boldsymbol{x}_n \right] = \boldsymbol{\Sigma}_N \left[\beta \mathbf{X}^\top \boldsymbol{y} \right] = (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta} \mathbf{I}_D)^{-1} \mathbf{X}^\top \boldsymbol{y} \quad \text{(posterior's mean)}
\end{aligned}$$

The Posterior: A Visualization

- Assume a linear regression problem with ground truth $\boldsymbol{w} = [w_0, w_1]$ with $w_0 = -0.3, w_1 = 0.5$
- Assume data generated by a linear regression model $y = w_0 + w_1 x +$ "noise"
  - Note: It's actually 1-D regression (w_0 is just a bias term), or 2-D reg. with feature $[1, x]$
- Figures below show the "data space" and posterior of $\boldsymbol{w}$ for different number of observations (note: with no observations, the posterior = prior)



- The "data space" (red lines) shown above denotes various possible linear regression datasets with data of the form $y = w_0 + w_1 x$ generated using $\boldsymbol{w}$ drawn from the current posterior of $\boldsymbol{w}$

Bayesian Linear Regression: Posterior Predictive Distribution

- Given the posterior $p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$, how to make prediction y_* for a new input $\boldsymbol{x}_*$?
- The posterior predictive distribution will be

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)d\boldsymbol{w}$$

- Using Gaussian predictive/marginal formula, the posterior predictive will be another Gaussian

$$\boxed{p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \mathcal{N}(\boldsymbol{\mu}_N^\top \boldsymbol{x}_*, \beta^{-1} + \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_*)}$$

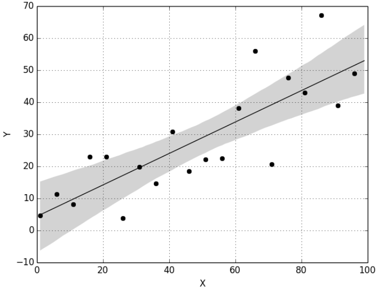
- So we get a predictive mean $\boldsymbol{\mu}_N^\top \boldsymbol{x}_*$ and an input-specific predictive variance $\beta^{-1} + \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_*$
- In contrast, MLE and MAP make "plug-in" predictions (using the point estimate of $\boldsymbol{w}$)

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \mathcal{N}(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*, \beta^{-1}) && \text{- MLE prediction} \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \mathcal{N}(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*, \beta^{-1}) && \text{- MAP prediction} \end{aligned}$$

- Important: Unlike MLE/MAP, the variance of y_* also depends on the input $\boldsymbol{x}_*$ (this, as we will see later, will be very useful in sequential decision-making problems such as active learning)

Posterior Predictive Distribution: An Illustration

Black dots are training examples



Width of the shaded region at any x denotes the predictive uncertainty at that x (+/- one std-dev)

Regions with more training examples have smaller predictive variance

Interpreting the Predictions

- We already saw that the posterior predictive for a new test point $\boldsymbol{x}_*$ is a Gaussian

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \mathcal{N}(\boldsymbol{\mu}_N^\top \boldsymbol{x}_*, \beta^{-1} + \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_*)$$

- (Exercise) Can show that the mean prediction at a new test input $\boldsymbol{x}_*$ can also be written as

$$\mathbb{E}_{p(\boldsymbol{w}|\boldsymbol{y},\mathbf{X})}[\boldsymbol{w}^\top \boldsymbol{x}_*] = \boldsymbol{\mu}_N^\top \boldsymbol{x}_* \quad = \quad \sum_{n=1}^{N} k(\boldsymbol{x}_n, \boldsymbol{x}_*) y_n \quad (\text{where } k(\boldsymbol{x}_n, \boldsymbol{x}_*) = \beta \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_n)$$

  .. which makes intuitive sense

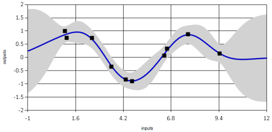
- (Exercise) Covariance of the predictions at any two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ will be

$$\text{cov}[\boldsymbol{x}^\top \boldsymbol{w}, \boldsymbol{w}^\top \boldsymbol{x}'] = \boldsymbol{x}^\top \boldsymbol{\Sigma}_N \boldsymbol{x}' = \beta^{-1} k(\boldsymbol{x}, \boldsymbol{x}')$$

  .. which makes intuitive sense (predictions of two faraway points will have low covariance)

- These interpretations will be useful when talking about Gaussian Processes for nonlinear regression

Nonlinear Regression?



- Can extend the linear regression model to handle nonlinear regression problems
- One way is to replace the feature vectors $\boldsymbol{x}$ by a nonlinear mapping $\phi(\boldsymbol{x})$

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}^\top \phi(\boldsymbol{x}), \beta^{-1})$$

- The nonlinear mapping can be defined directly, e.g., for a one-dimensional feature x

$$\phi(x) = [1, x, x^2]$$

- Alternatively, a kernel function can be used to <u>implicitly</u> define the nonlinear mapping
- More on nonlinear regression when we discuss Gaussian Processes

What about the hyperparameters of the regression model?

- If hyperparameters are to be estimated, we will have a hierarchical/multiparameter model
- Posterior inference in slightly more involved in this case
- Iterative methods required to learn the weight vector and the hyperparameters, e.g.,
  - Marginal likelihood maximization for hyperparameter estimation
  - Expectation maximization (EM)
  - MCMC or variational inference
- We will discuss more when we talk about inference in hierarchical/multiparameter models