Data Analytics
EEE 4774 & 6777

Module 5 - Regression

Linear Regression

Spring 2022

Linear Regression

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \cdots + w_M x_M$$
$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}$$

- Input variables (regressors, independent variables, predictors, features): $\boldsymbol{x}$
- Output variables (dependent variables, targets): y
- Unknown parameters (regression coefficients): $\boldsymbol{w}$

Maximum Likelihood and Least Squares

- Assume observations from a deterministic function with added Gaussian noise:

$$t_n = \boldsymbol{w}^T \boldsymbol{x}_n + z_n \quad \text{where} \quad z_n \sim \mathcal{N}(0, \beta^{-1})$$

which is the same as saying,

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

- Given observed inputs, $\boldsymbol{X}^{N \times M}$, and targets, $\boldsymbol{t} = [t_1, \dots, t_N]^T$ we obtain the likelihood function

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T \boldsymbol{x}_n, \beta^{-1})$$

Maximum Likelihood and Least Squares

$$t_n = w^T x_n + z_n$$

Typically, standard Gaussian with $\beta = 1$

$$\log p(t|w,\beta) = \sum_{n=1}^{N} \log \mathcal{N}(t_n | w^T x_n, \beta^{-1})$$

ML

Sum-of-squares error $E_D(\boldsymbol{w})$

$$w_{ML} = \arg\min_{w} \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T x_n)^2$$

LS

Computing the gradient and setting it to zero yields

$$w_{ML} = \left(\sum_{n=1}^{N} x_n x_n^T \right)^{-1} \sum_{n=1}^{N} t_n x_n$$

ML=LS for the Gaussian case

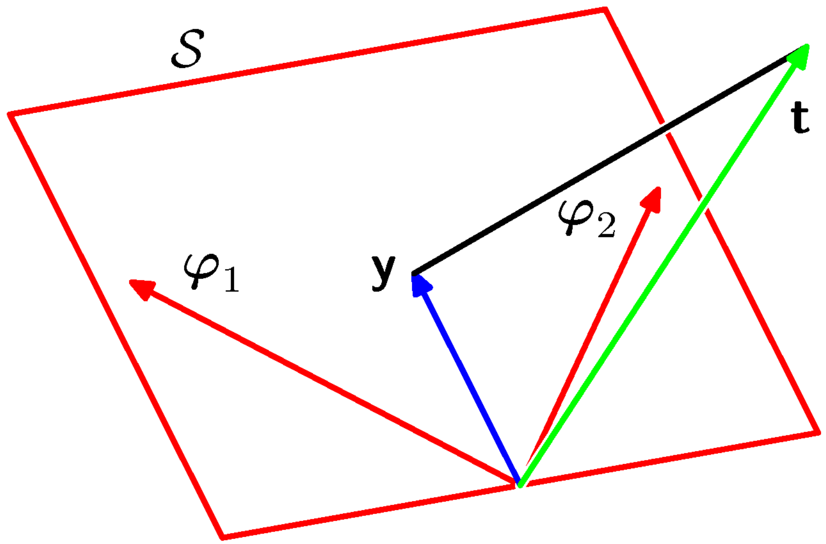
$$w_{ML} = \left(X^T X \right)^{-1} X^T t$$

Geometry of Least Squares

- Consider

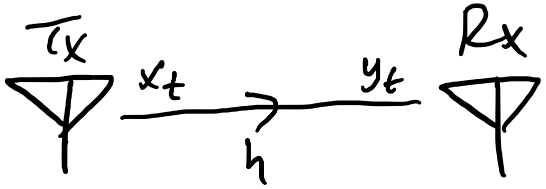
$$\boldsymbol{y} = X\boldsymbol{w}_{ML} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_M]\boldsymbol{w}_{ML}$$

$$\mathbf{y} \in \underbrace{\mathcal{S}}_{\text{M-dimensional}} \subseteq \underbrace{\mathcal{T}}_{\text{N-dimensional}} \qquad \mathbf{t} \in \mathcal{T}$$



- S is spanned by $\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_M$.
- $\boldsymbol{w}_{ML}$ minimizes the distance between t and its orthogonal projection on S, i.e. y.

Linear Regression Example:
Channel Estimation in Wireless Communications



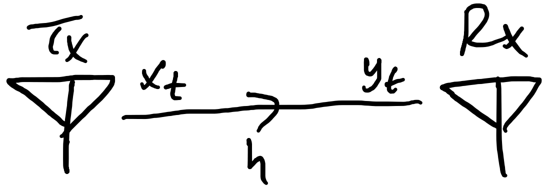
$$y_t = h x_t + z_t$$

received, channel coefficient, transmitted, noise

$$z_t \sim \mathcal{N}(0, \sigma^2)$$

Given $\{x_t, y_t\}_t$ estimate h

Linear Regression Example:
Channel Estimation in Wireless Communications



$$y_t = h x_t + z_t$$

received, channel coefficient, transmitted, noise

$$z_t \sim \mathcal{N}(0, \sigma^2)$$

Given $\{x_t, y_t\}_t$ estimate h

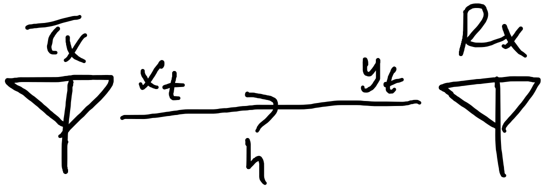
Least Squares Estimation

$$\hat{h}_{LS} = \arg\min_{h} \sum_{t=1}^{T} (y_t - h x_t)^2$$

$$\frac{\partial}{\partial h} \sum_{t=1}^{T} (y_t - h x_t)^2 \Big|_{h = \hat{h}_{LS}} = \sum_{t=1}^{T} 2(y_t - \hat{h}_{LS} x_t)(-x_t) = 0 \Rightarrow \boxed{\hat{h}_{LS} = \frac{\sum_{t=1}^{T} y_t x_t}{\sum_{t=1}^{T} x_t^2}}$$

Linear Regression Example:

Channel Estimation in Wireless Communications



$$y_t = h x_t + z_t$$

received, channel coefficient, transmitted, noise

$$z_t \sim \mathcal{N}(0, \sigma^2)$$

Given $\{x_t, y_t\}_t$ estimate h

Least Squares Estimation

$$\hat{h}_{LS} = \arg\min_h \sum_{t=1}^{T} (y_t - h x_t)^2$$

$$\frac{\partial}{\partial h} \sum_{t=1}^{T} (y_t - h x_t)^2 \Big|_{h=\hat{h}_{LS}} = \sum_{t=1}^{T} 2(y_t - \hat{h}_{LS} x_t)(-x_t) = 0 \Rightarrow \boxed{\hat{h}_{LS} = \frac{\sum_{t=1}^{T} y_t x_t}{\sum_{t=1}^{T} x_t^2}}$$

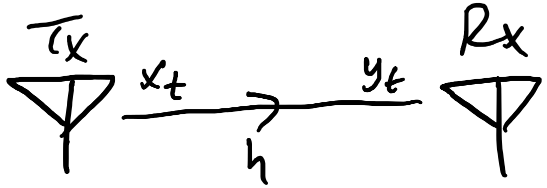
ML Estimation

$$\hat{h}_{ML} = \arg\max_h \log \frac{e^{-\sum_{t=1}^{T} \frac{(y_t - h x_t)^2}{2\sigma^2}}}{(2\pi\sigma^2)^{T/2}}$$

$$\hat{h}_{ML} = \arg\min_h \sum_{t=1}^{T} (y_t - h x_t)^2$$

$$\boxed{\hat{h}_{ML} = \hat{h}_{LS}}$$

Linear Regression Example:
Channel Estimation in Wireless Communications

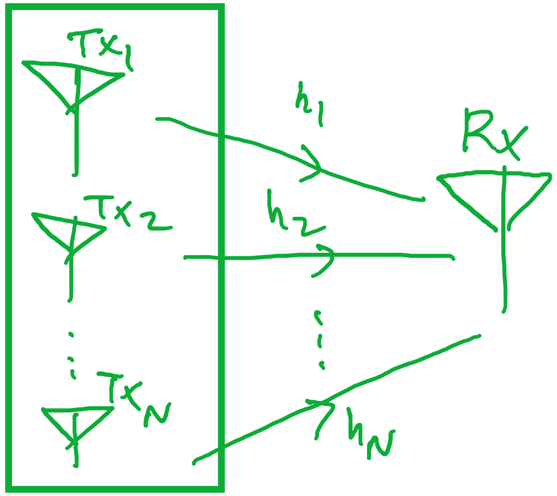


$$y_t = h x_t + z_t$$

received channel coefficient transmitted noise

$$z_t \sim \mathcal{N}(0, \sigma^2)$$

Given $\{x_t, y_t\}_t$ estimate h



MISO : $y_t = X_t' H + z_t$

$$\hat{H}_{LS} = \arg\min_{H} \sum_{t=1}^{T} (y_t - X_t' H)^2$$

$$\frac{\partial}{\partial H}$$

$$\sum_{t=1}^{T} 2(y_t - X_t' \hat{H}_{LS})(-X_t) = 0$$

$$\sum_{t=1}^{T} (X_t X_t') \hat{H}_{LS} = \sum_{t=1}^{T} X_t y_t$$

$$X^{T\times N} = \begin{bmatrix} X_1' \\ \vdots \\ X_T' \end{bmatrix}$$

Least Squares Estimate

$$\boxed{\hat{H}_{LS} = (X^T X)^{-1} X^T Y}$$

$$Y = XH + Z$$

Least Squares Estimation

$$\hat{h}_{LS} = \arg\min_{h} \sum_{t=1}^{T} (y_t - h x_t)^2$$

$$\frac{\partial}{\partial h} \sum_{t=1}^{T} (y_t - h x_t)^2 \Big|_{h = \hat{h}_{LS}} = \sum_{t=1}^{T} 2(y_t - \hat{h}_{LS} x_t)(-x_t) = 0 \Rightarrow \boxed{\hat{h}_{LS} = \frac{\sum_{t=1}^{T} y_t x_t}{\sum_{t=1}^{T} x_t^2}}$$

ML Estimation

$$\hat{h}_{ML} = \arg\max_{h} \log \frac{e^{-\frac{\sum_{t=1}^{T} (y_t - h x_t)^2}{2\sigma^2}}}{(2\pi\sigma^2)^{T/2}}$$

$$\hat{h}_{ML} = \arg\min_{h} \sum_{t=1}^{T} (y_t - h x_t)^2$$

$$\boxed{\hat{h}_{ML} = \hat{h}_{LS}}$$

Linear Regression Example:
Autoregressive (AR) Model

- AR(p): the next value depends on the previous p values

$$x_t = w_0 + w_1 x_{t-1} + w_2 x_{t-2} + \cdots + w_p x_{t-p} + z_t$$

$$x_t = w_0 + \sum_{i=1}^{p} w_i x_{t-i} + z_t$$

Regularized Least Squares

- Consider the error function:

$$\beta E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

- With the sum-of-squares error function and a quadratic regularizer, we get

$$\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- which is minimized by

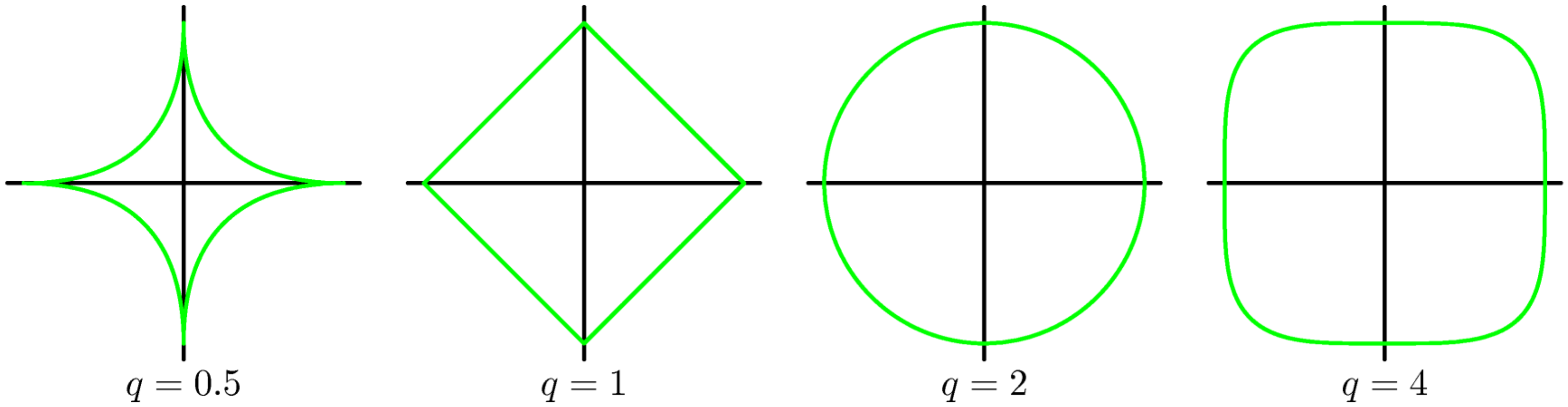
$$\mathbf{w} = \left(\frac{\lambda \mathbf{I}}{\beta} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

λ is called the regularization coefficient.

Regularized Least Squares

- With a more general regularizer, we have

$$\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

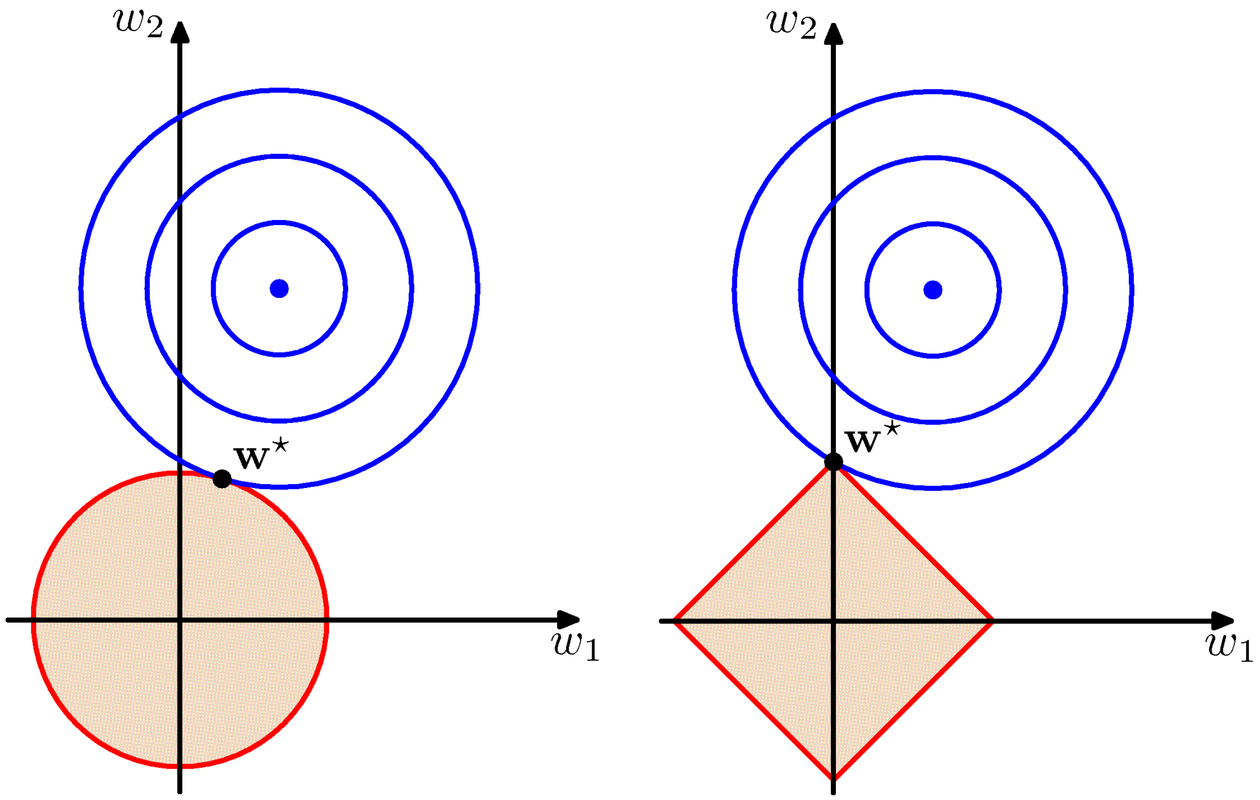


Lasso

Quadratic

Regularized Least Squares

- Lasso tends to generate sparser solutions than a quadratic regularizer.



How to select regularization coefficient λ?
The Bias-Variance Decomposition

- Recall the *expected squared loss, i.e., Mean Squared Loss (MSE)*,

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

where

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

- The second term of E[L] corresponds to the noise inherent in the random variable t.

- What about the first term?

The Bias-Variance Decomposition

- Suppose we were given multiple data sets, each of size N. Any particular data set, D, will give a particular function y(x;D). We then have

$$
\begin{aligned}
&\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 \\
&\quad = \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad = \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\qquad +2\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}.
\end{aligned}
$$

The Bias-Variance Decomposition

- Taking the expectation over D yields

$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right]$$
$$= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.$$

The Bias-Variance Decomposition

- Thus we can write

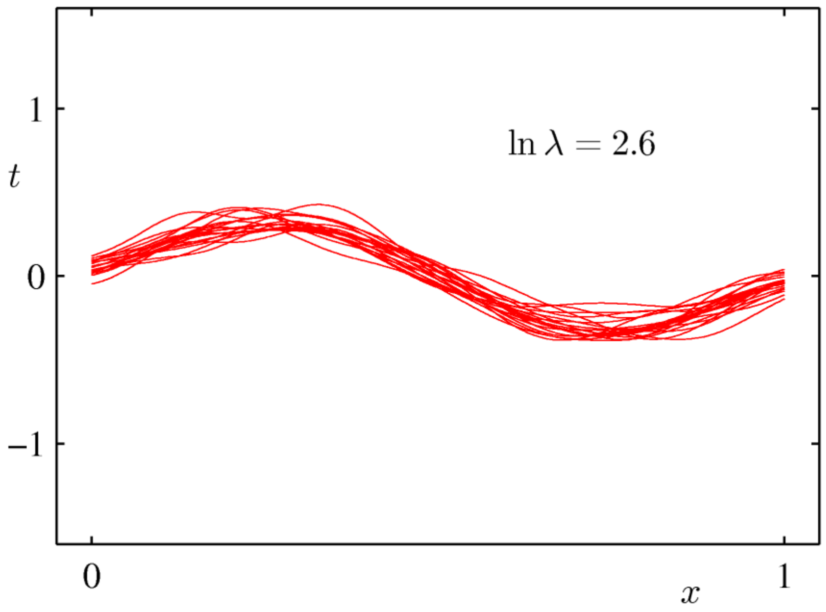
$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

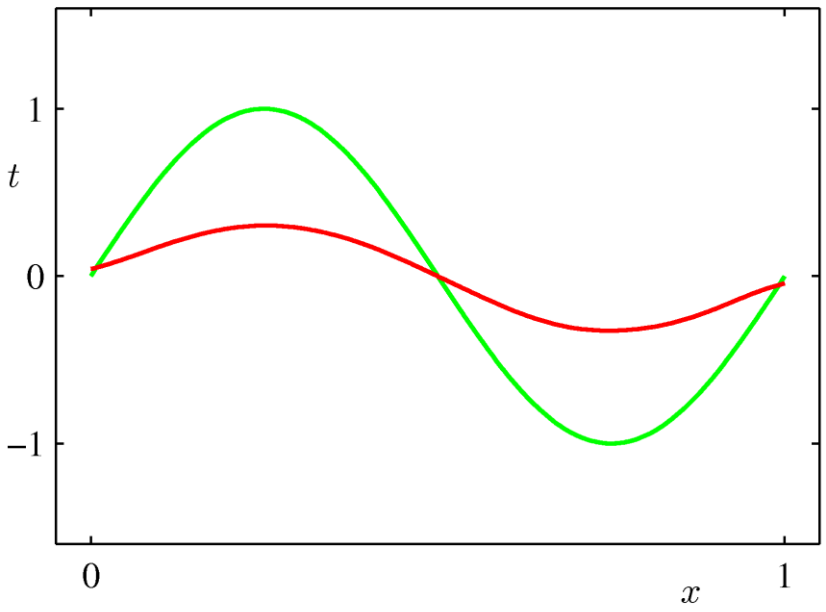
where

$$\begin{aligned}
(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{noise} &= \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t
\end{aligned}$$

The Bias-Variance Decomposition

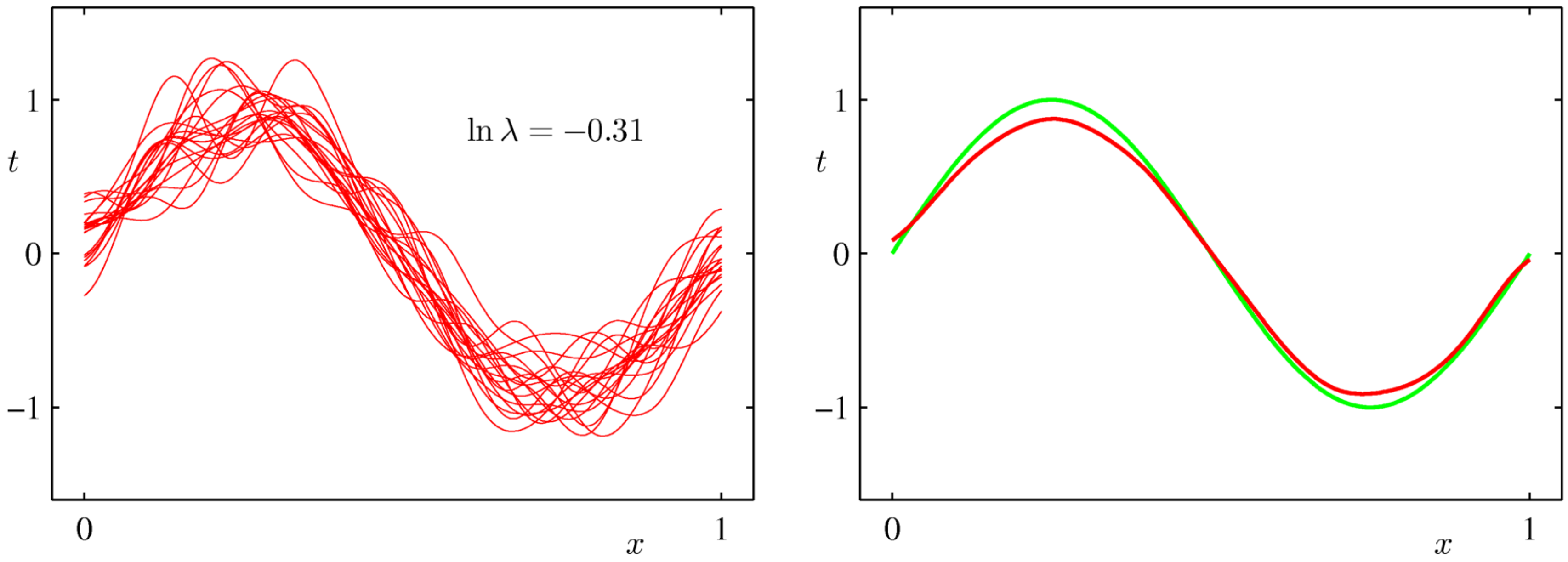
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.





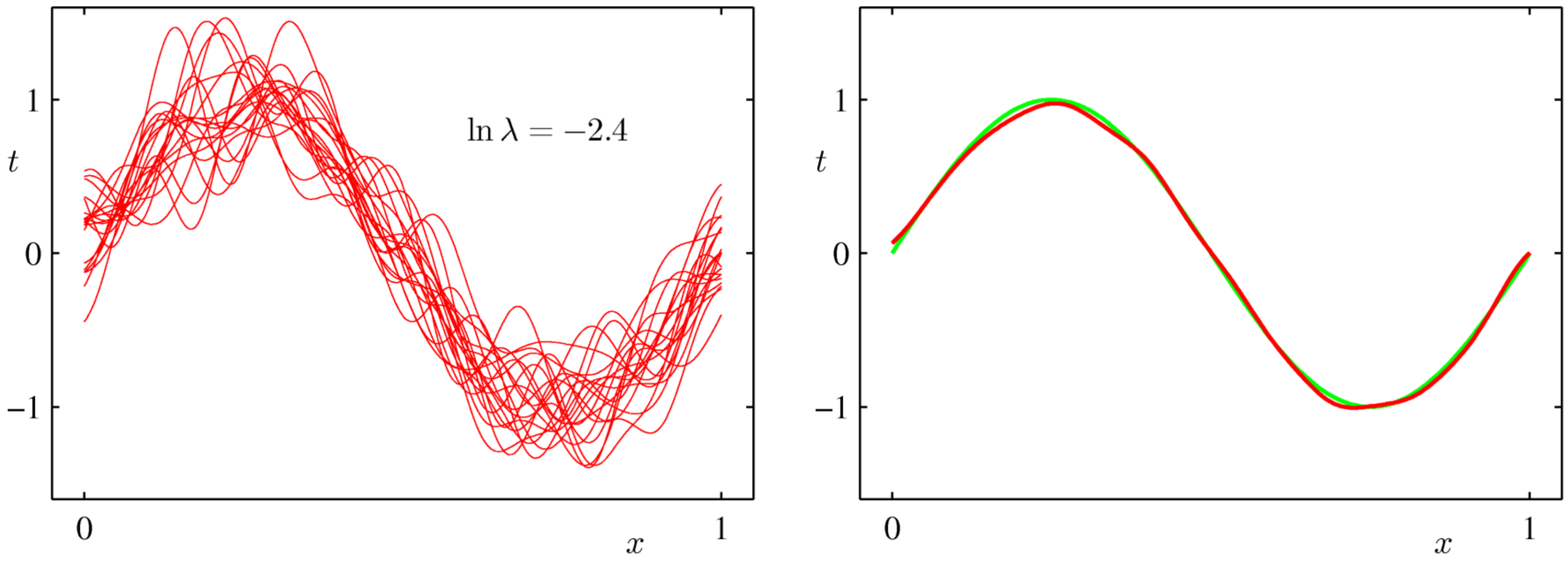
The Bias-Variance Decomposition

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.



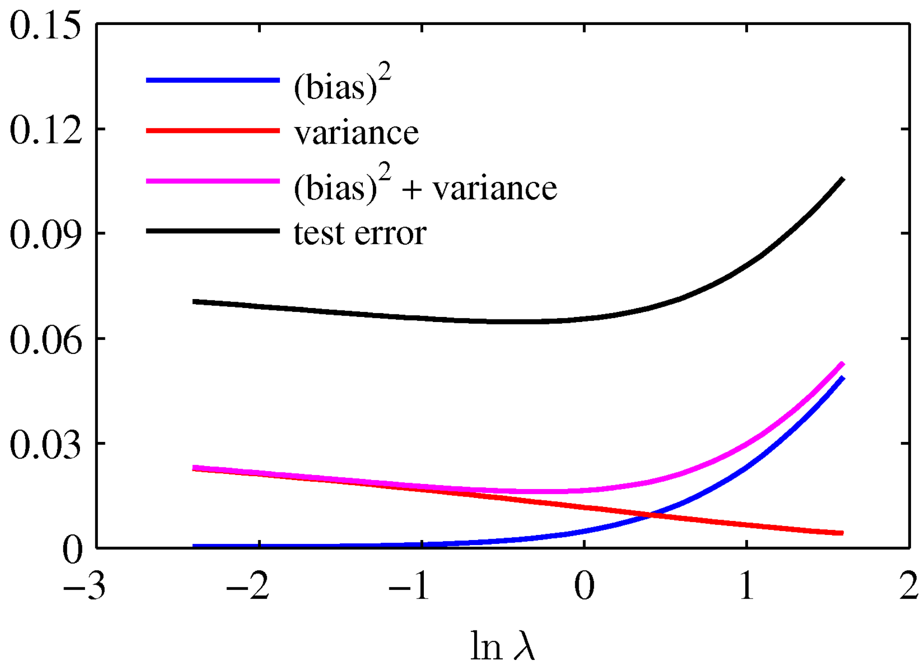
The Bias-Variance Decomposition

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ.



The Bias-Variance Trade-off

- From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance.



Bayesian Linear Regression

- Define a conjugate prior over w

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0).$$

•Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

- where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}$$

Bayesian Linear Regression

- A common choice for the prior is

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

- for which

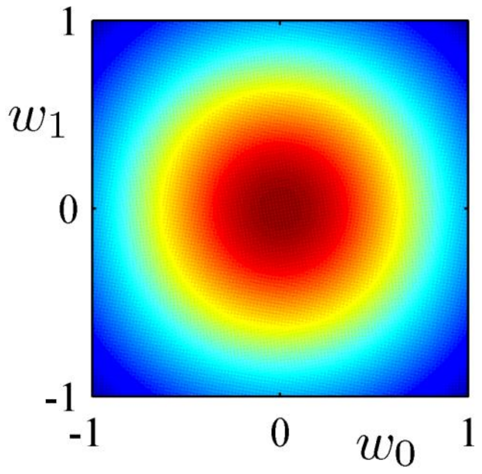
$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}. \end{aligned}$$

- Next we consider an example …

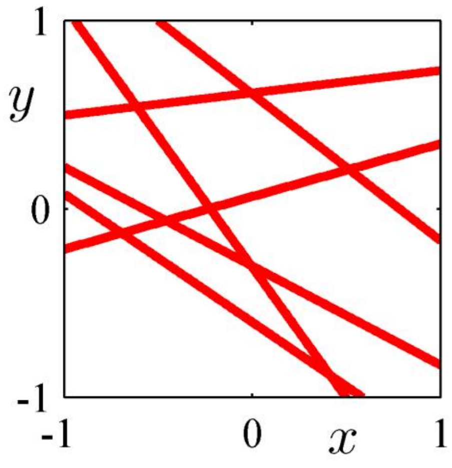
Bayesian Linear Regression

0 data points observed

Prior



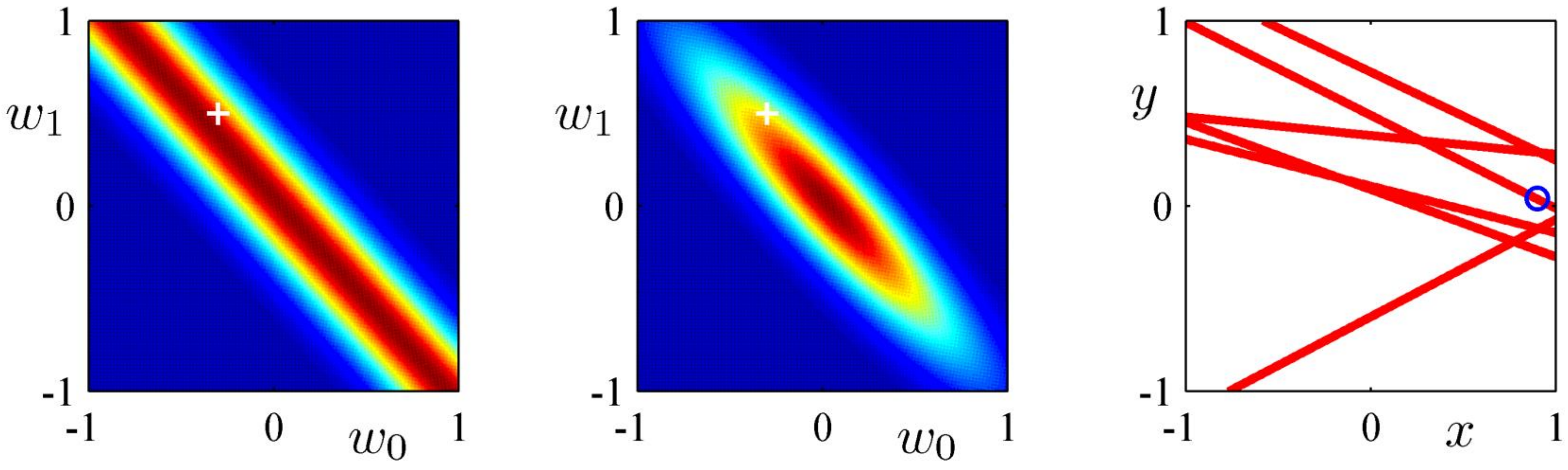
Data Space: samples from w_0 and w_1



Bayesian Linear Regression

1 data point observed

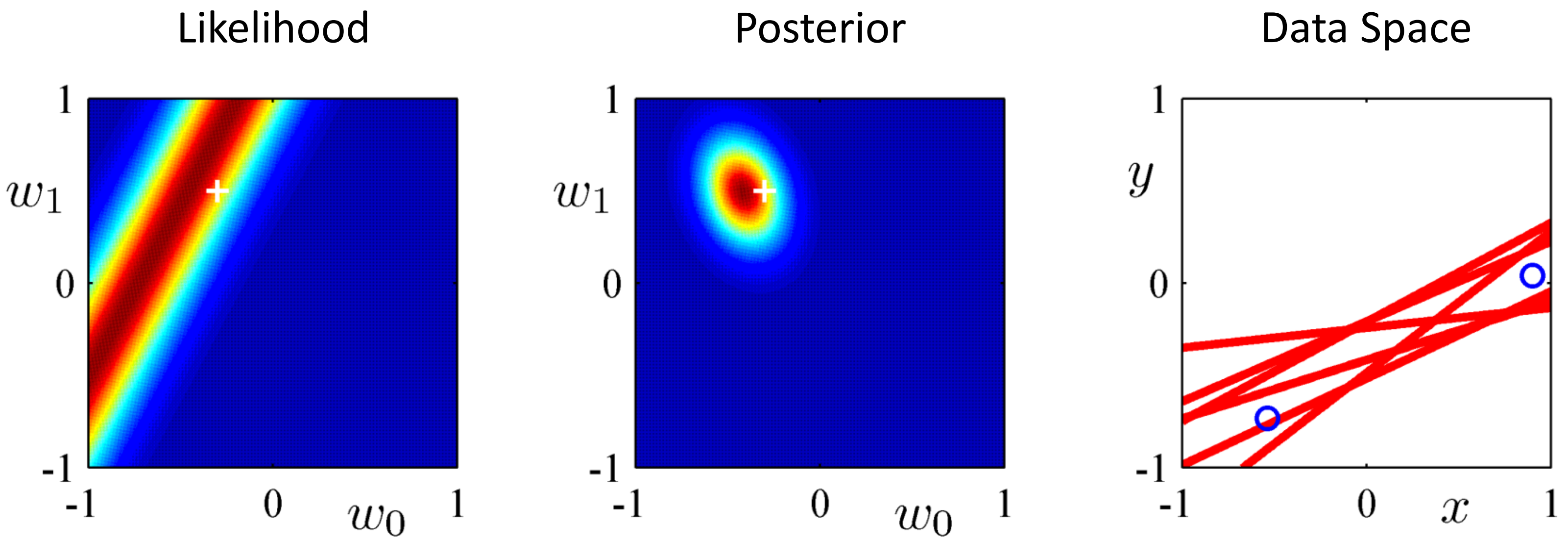
Likelihood | Posterior | Data Space



Likelihood . Prior

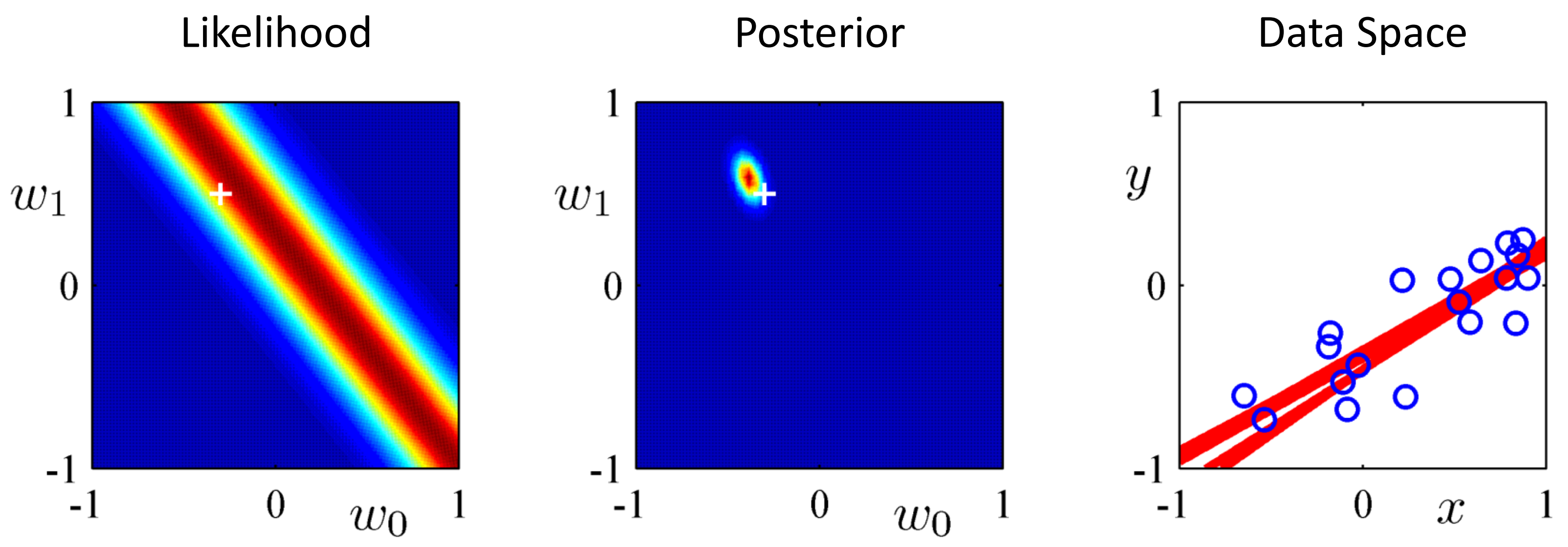
Bayesian Linear Regression

2 data points observed



Bayesian Linear Regression

20 data points observed



Sequential Learning

- Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

$$\begin{aligned}\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n))\boldsymbol{\phi}(\mathbf{x}_n).\end{aligned}$$

- This is known as the *least-mean-squares (LMS) algorithm*. Issue: how to choose η?

Sequential Learning

$MSE = E[(t_n - W^T\phi(x_n))^2]$, $\hat{W}_{MSE} = \arg\min_{W} MSE(w)$

$\hat{W}_{LS} = \arg\min_{W} \sum_{n=1}^{N} (t_n - W^T\phi(x_n))^2$

$\Rightarrow \hat{W}_{MSE} = E[\phi\phi^T]^{-1} E[\phi^T t]$

$= \underbrace{(\sum_n \phi\phi^T)^{-1} \sum_n \phi t_n}_{\text{Batch Learning}}$

estimates

- Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

Gradient Descent : Iterative optimization — move towards negative of gradiant of $\underbrace{\text{cost function}}_{\text{MSE}}$

$E[\nabla_w (t_n - w^T\phi)^2]$

$= \underbrace{E[2(t_n - w^T\phi)(-\phi)]}_{\text{Difficult to compute online}}$

can be estimated by

$\frac{1}{N}\sum_{n=1}^{N} (t_n - w^T\phi)\phi$

Stochastic Gradient Descent: Use single-point estimate $\boxed{(t_n - w^T\phi)\phi}$

$$
\begin{aligned}
\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\
&= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\phi(\mathbf{x}_n))\phi(\mathbf{x}_n).
\end{aligned}
$$

- This is known as the *least-mean-squares (LMS) algorithm*. Issue: how to choose η?