

Linear Regression

Rob Weston

Applied Artificial Intelligence Lab

March 23, 2021

Overview

1. Linear Regression

2. Frequentist Linear Regression

Linear Regression - A frequentist approach

As optimal function estimation

Learning from noisy observations

Overfitting and problems with finite datasets

Regularisation

3. Bayesian Linear Regression

Our example revisited

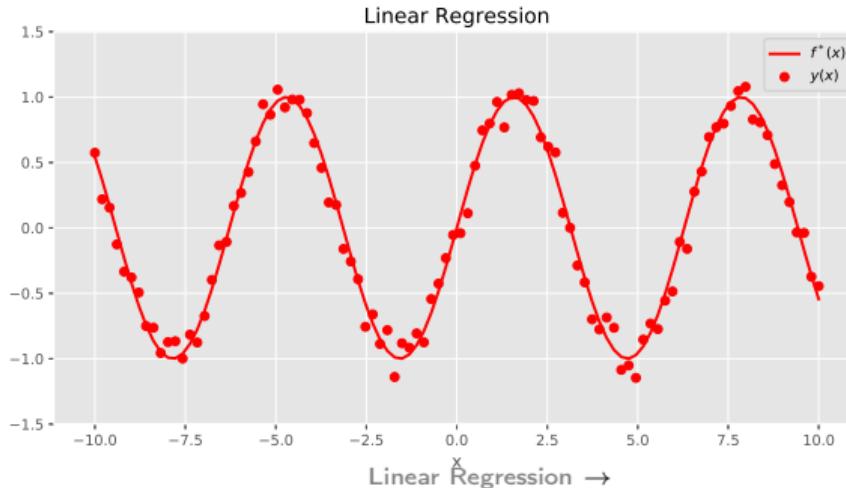
The Effective number of Parameters

Linear Regression

Estimate a function $f^* : \mathbb{R}^D \rightarrow \mathbb{R}$ with a linear model $f(x) = \mathbf{w}^\top \phi(x)$ from observations $y(x) = f^*(x) + \epsilon$ corrupted by noise $\epsilon \sim \text{Norm}(0, \alpha^{-1})$ where...

- $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$ is a *known* function mapping $x \in \mathbb{R}^D$ to a feature space $\phi(x) \in \mathbb{R}^M$.
- $\mathbf{w} \in \mathbb{R}^M$ is used to weight each feature in ϕ
- α is the *observation precision* and is assumed a hyper-parameter

The *bias* is assumed to be included in \mathbf{w} corresponding to a fixed feature $\phi(x) = 1$. Code for the example below can be found [here](#).



Overview

1. Linear Regression

2. Frequentist Linear Regression

Linear Regression - A frequentist approach

As optimal function estimation

Learning from noisy observations

Overfitting and problems with finite datasets

Regularisation

3. Bayesian Linear Regression

Our example revisited

The Effective number of Parameters

Linear Regression - A frequentist approach

- Find the weights that maximise the likelihood $\mathbf{w}^* = \arg \max_{\mathbf{w}} \log p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \alpha)$ given N observations $\mathbf{y} = [y_1, \dots, y_N] \in \mathbb{R}^N$ where

$$\log p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \alpha) = \log \prod_{n=1}^N \text{Nor}(y_n | \mathbf{w}^\top \boldsymbol{\phi}_n, \alpha^{-1}) = \frac{N}{2} \log \alpha - \frac{N}{2} \log 2\pi - \sum_{n=1}^N (y_n - \mathbf{w}^\top \boldsymbol{\phi}_n)^2 = -E(\mathbf{w})$$

- Differentiating with respect to \mathbf{w} and setting to 0 gives $\mathbf{w}^* = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$ is equal to the *Moore-Penrose Inverse* where and $\Phi = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_M)]^\top \in \mathbb{R}^{N \times M}$ is the *design matrix*.
- Similarly for α we have $(\alpha^*)^{-1} = \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w}^{*\top} \phi(\mathbf{x}_n))^2$
- Assuming we have inputs $\mathbf{x} \sim p(\mathbf{x})$ and letting $N \rightarrow \infty$, optimising over \mathbf{w} is equivalent to¹

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [(y(\mathbf{x}) - \mathbf{w}^\top \phi(\mathbf{x}))^2] = \arg \min_{\mathbf{w}} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\ell(y(\mathbf{x}), f(\mathbf{x}; \mathbf{w}))]$$

¹This comes from the fact that \mathbf{w} only depends on the term $\sum_{n=1}^N (y_n - \mathbf{w}^\top \boldsymbol{\phi}_n)^2$ in $E(\mathbf{w})$

Linear Regression as Optimal Function Estimation

Instead of viewing the optimisation over parameters w we can view the optimisation over functions f

- We aim to estimate a function f^* from observations $y(x) = f^*(x) + \epsilon$ corrupted by noise ϵ . We have

$$\hat{f} = \arg \min_f \left\{ \mathbb{E}_{p(x,y)} [\ell(y(x), f(x))] \right\} = \arg \min_f \{L[f]\}$$

where $\ell(y(x), f(x))$ is a loss function between points $y(x)$ and $f(x)$.

- This can be viewed as a *calculus of variations* problem \Rightarrow "determine the *function* $f(x)$ which minimises the *functional* $L[f] = \mathbb{E}_{p(x,y)} [\ell(y(x), f(x))]$ ".
- The optimum estimator \hat{f} is determined by our *choice* of loss function ℓ .
- For $\ell(y(x), f(x)) = (y(x) - f(x))^2$ it can be shown that $\hat{f}(x) = \mathbb{E}_{p(y|x)}[y(x)]$ and so $\hat{f}(x) = f^*(x)$ (as the mean of $p(y|x) = \mathcal{N}(y(x)|f^*(x), \sigma^2)$ is $f^*(x)$).
- In this case $L[f]$ becomes the *mean squared error* Mse[f]

Learning from Noisy Observations

Maximum Likelihood with Noisy Observations

When we estimate $f(\mathbf{x})$ from noisy observations $y(\mathbf{x}) = f^*(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ using maximum likelihood we are guaranteed to converge to the true solution $f(\mathbf{x}) = f^*(\mathbf{x})$ corresponding to an optimal loss

$$\text{Mse}[f] = \mathbb{E}_{p(\mathbf{x}, y)}[(y(\mathbf{x}) - f(\mathbf{x}))^2] = \sigma^2$$

Proof Writing $y = y(\mathbf{x})$, $f = f(\mathbf{x})$ and $f^* = f^*(\mathbf{x})$

$$\text{Mse}[f] = \mathbb{E}_{p(\mathbf{x}, y)}[(y - f)^2] \tag{1}$$

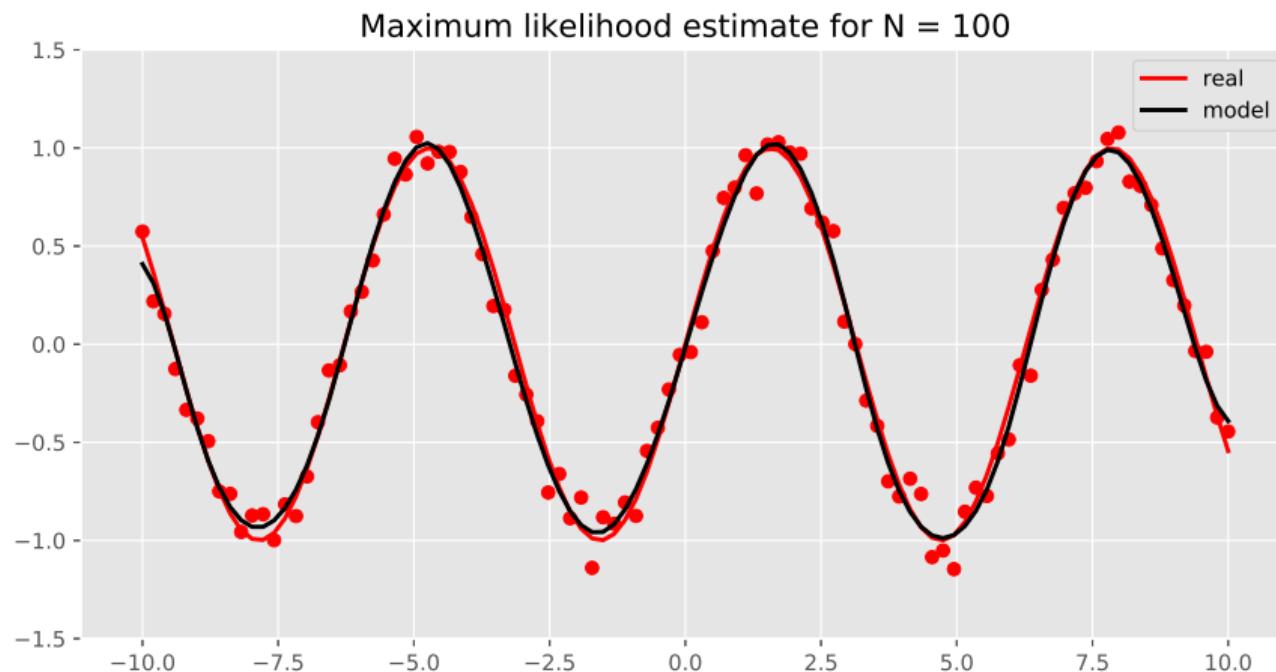
$$= \mathbb{E}_{p(\mathbf{x}, y)}[(y - f^* + f^* - f)^2] \tag{2}$$

$$= \mathbb{E}_{p(\mathbf{x}, y)}[(y - f^*)^2] + \mathbb{E}_{p(\mathbf{x}, y)}[(f^* - f)^2] + 2\mathbb{E}_{p(\mathbf{x}, y)}[(f^* - f)(y - f^*)] \tag{3}$$

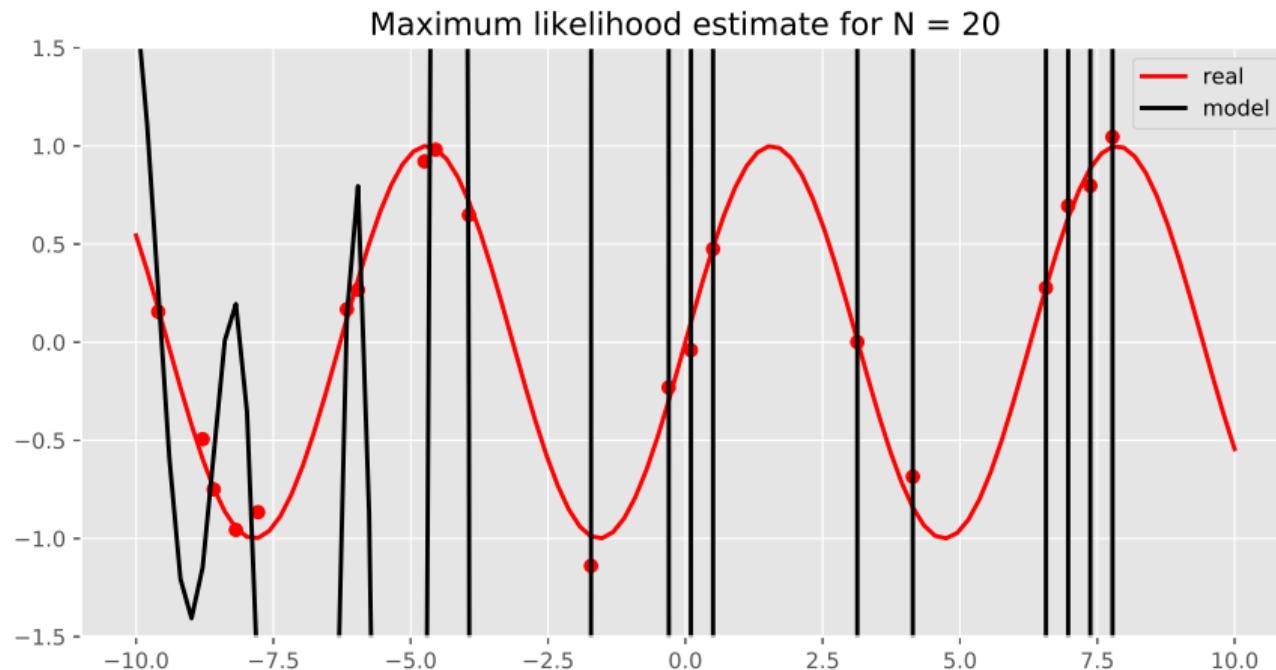
$$= \mathbb{E}_{p(\mathbf{x}, y)}[(y - f^*)^2] + \mathbb{E}_{p(\mathbf{x}, y)}[(f^* - f)^2] \quad \text{assuming } (f^* - f) \text{ and } (y - f^*) \text{ are uncorrelated} \tag{4}$$

If we have $\mathbb{E}[\epsilon] = 0$ then $\mathbb{E}_{p(y|\mathbf{x})}[y] = \mathbb{E}_{p(y|\mathbf{x})}[f^*(\mathbf{x}) + \epsilon] = f^*(\mathbf{x})$. The expected loss in this case is given as $\text{Mse}[f] = \mathbb{E}_{p(\mathbf{x}, y)}[(y(\mathbf{x}) - f^*(\mathbf{x}))^2] = \mathbb{E}_{p(\mathbf{x}, y)}[\epsilon^2] = \sigma^2$.

Maximum Likelihood is good when N is big...



... but bad when N is small!



Why?

Working with Finite datasets

When we work with a *finite* dataset \mathcal{D} we are no longer estimating the *ideal* maximum likelihood estimate $\hat{f}(x)$ but instead approximating it with $\hat{f}(x) \approx \tilde{f}(x; \mathcal{D})$. Different datasets $\mathcal{D} \sim p(\mathcal{D})$ will result in different estimates $\tilde{f}(x; \mathcal{D})$. In this case our estimate \tilde{f} is no longer ideal and the expected mean squared error becomes

$$\text{Mse}[\tilde{f}] = \sigma^2 + \text{Bias}^2[\tilde{f}] + \text{Var}[\tilde{f}]$$

where σ^2 is the *observation* variance, $\text{Bias}[\tilde{f}] = \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])^2\}$ is the bias in the estimator \tilde{f} and $\text{Var}[\tilde{f}] = \mathbb{E}_{p(x)}\{(\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2\}$ is the variance in \tilde{f} .

Proof As before we have $L[\tilde{f}] = \mathbb{E}_{p(x,y)}[(y - f^*)^2] + \mathbb{E}_{p(x)}[(f^* - \tilde{f})^2]$. Expanding the second term gives,

$$\mathbb{E}_{p(x)}\{(f^* - \tilde{f})^2\} = \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}] + \mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2\} \quad (5)$$

$$= \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])^2 + (\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2 + 2(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])(\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})\} \quad (6)$$

$$= \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])^2\} + \mathbb{E}_{p(x)}\{(\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2\} \quad (7)$$

$$= \text{Bias}^2(\tilde{f}) + \text{Var}(\tilde{f}) \quad (8)$$

Noting that $\mathbb{E}_{p(x,y)}[(y - f^*)^2] = \sigma^2$ gives $L[\tilde{f}] = \sigma^2 + \text{Bias}^2(\tilde{f}) + \text{Var}(\tilde{f})$.

Why?

Working with Finite datasets

When we work with a *finite* dataset \mathcal{D} we are no longer estimating the *ideal* maximum likelihood estimate $\hat{f}(x)$ but instead approximating it with $\hat{f}(x) \approx \tilde{f}(x; \mathcal{D})$. Different datasets $\mathcal{D} \sim p(\mathcal{D})$ will result in different estimates $\tilde{f}(x; \mathcal{D})$. In this case our estimate \tilde{f} is no longer ideal and the expected mean squared error becomes

$$\text{Mse}[\tilde{f}] = \sigma^2 + \text{Bias}^2[\tilde{f}] + \text{Var}[\tilde{f}]$$

where σ^2 is the *observation* variance, $\text{Bias}[\tilde{f}] = \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])^2\}$ is the bias in the estimator \tilde{f} and $\text{Var}[\tilde{f}] = \mathbb{E}_{p(x)}\{(\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2\}$ is the variance in \tilde{f} .

Note:

- In the limit as $N \rightarrow \infty$ we have $p(\mathcal{D}) = \delta(\mathcal{D} - \mathcal{D}^*)$ and $\tilde{f} \rightarrow \hat{f} = f^*(x)$: this is why maximum likelihood is good for large N!

Regularisation

From the perspective of Occams Razor

- When N is small the maximum likelihood solution *overfits* to the data (good at predicting training data but bad at generalising)
- Solution = *Occam's Razor* \implies "Prefer the simpler model"
- Could make the model simpler by reducing the dimensionality of the feature space M and so limiting the number of parameters $\mathbf{w} \in \mathbb{R}^M$ in our model. But difficult to optimise over...
- Instead penalise the norm of the weights $\|\mathbf{w}\|$ and consider $\mathbf{w}^* = \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$ with

$$\mathcal{L}(\mathbf{w}) = E(\mathbf{w}) + \lambda \|\mathbf{w}\|$$

where $E(\mathbf{w})$ is as defined before.

- For $\|\mathbf{w}\| = \|\mathbf{w}\|_2^2$ we can arrive at $\mathcal{L}(\mathbf{w})$ considering a MAP estimate for \mathbf{w} with a prior $p(\mathbf{w}) = \text{Norm}(\mathbf{w}|0, \beta^{-1} \mathbf{I})$ and in this case $\mathbf{w}^* = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$.
- Could also consider $\|\mathbf{w}\| = \|\mathbf{w}\|_1$. In this case the weights in \mathbf{w} are encouraged to be sparse (eg. identical to 0).

Regularisation

From the perspective of Bias and Variance

- The *expected mean squared error* for a non-ideal estimator is given as

$$\text{Mse}[\tilde{f}] = \sigma^2 + \text{Bias}^2[\tilde{f}] + \text{Var}[\tilde{f}]$$

where σ^2 is the *observation variance*, $\text{Bias}[\tilde{f}] = \mathbb{E}_{p(x)}\{(f^* - \mathbb{E}_{p(\mathcal{D})}[\tilde{f}])^2\}$ is the bias in the estimator \tilde{f} and $\text{Var}[\tilde{f}] = \mathbb{E}_{p(x)}\{(\mathbb{E}_{p(\mathcal{D})}[\tilde{f}] - \tilde{f})^2\}$ is the variance in \tilde{f} .

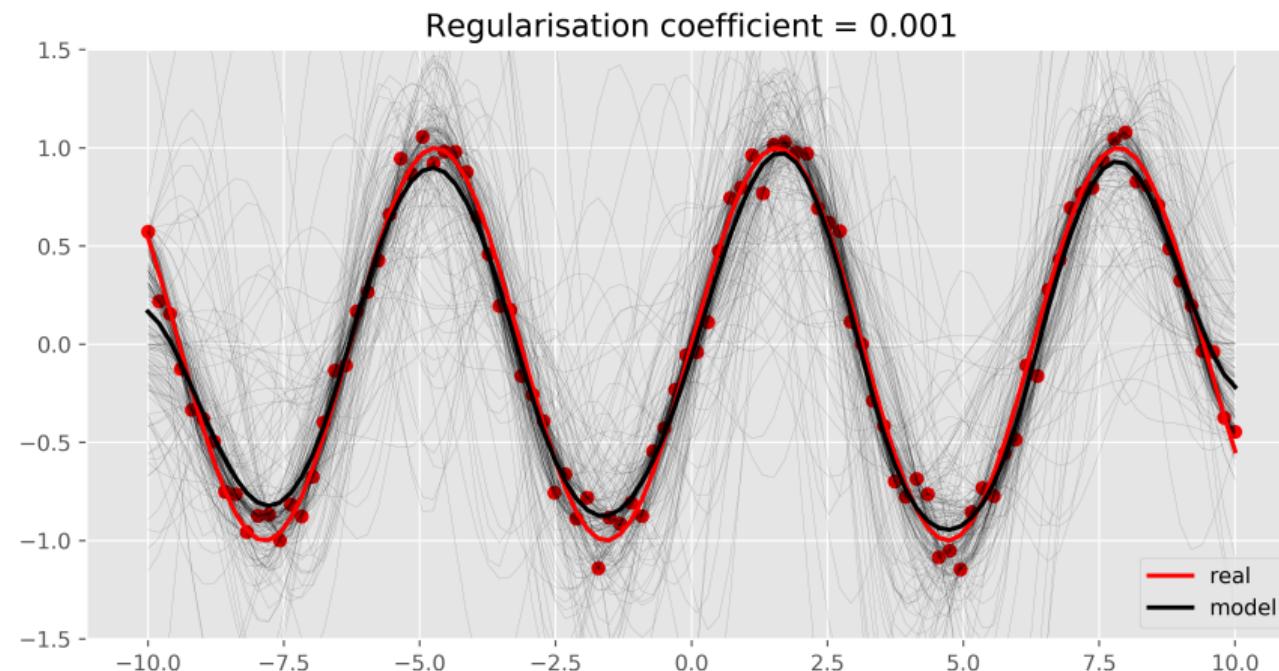
- Observation:* The maximum-likelihood estimate is just one of a host of estimates we could use for \tilde{f} and the above decomposition remains true independent of the loss function we actually optimise to find \tilde{f}
- Idea:* "design a new loss function \mathcal{L} which results in smaller mean squared error $L[\tilde{f}]$ when N is small"
- By introducing regularisation

$$\mathcal{L}(\mathbf{w}) = E(\mathbf{w}) + \lambda \|\mathbf{w}\|$$

we increase the $\text{Bias}[\tilde{f}]$ but *reduce* $\text{Var}[\tilde{f}]$ for small N . For a correct weighting λ this *can* result in smaller expected mean squared error $\text{Mse}[\tilde{f}] = \mathbb{E}_{p(x)}[(\tilde{f}(x) - f^*(x))^2]$

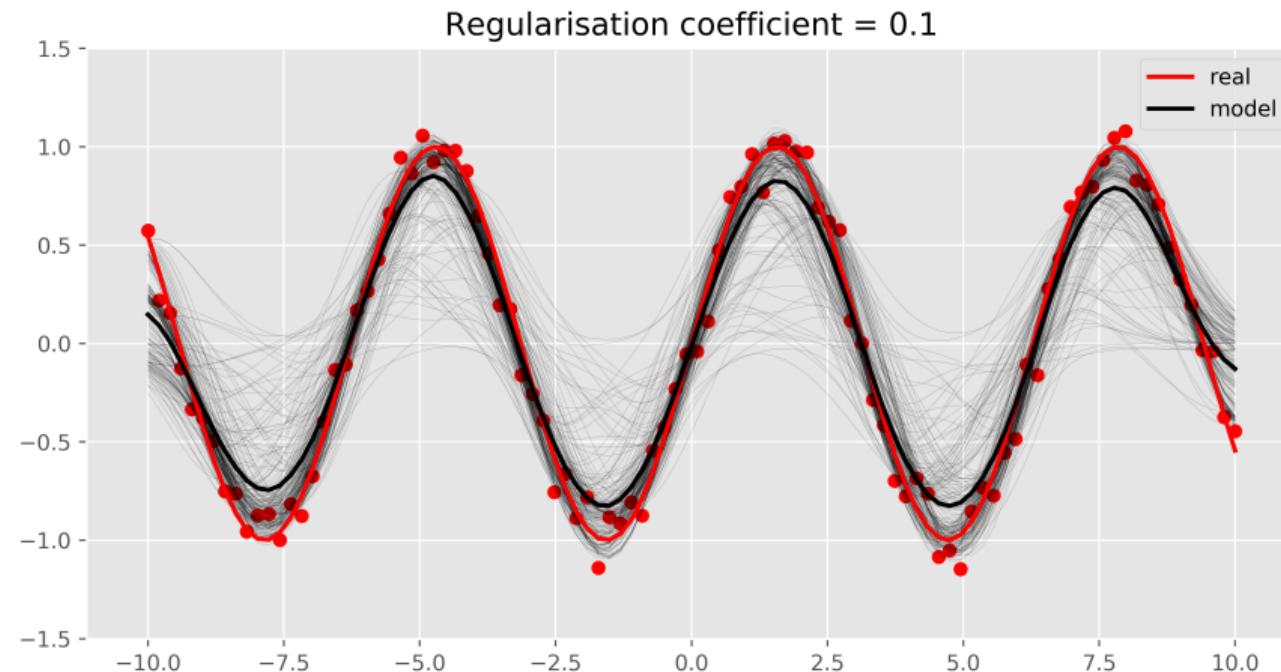
Regularisation

High Variance - Low Bias...



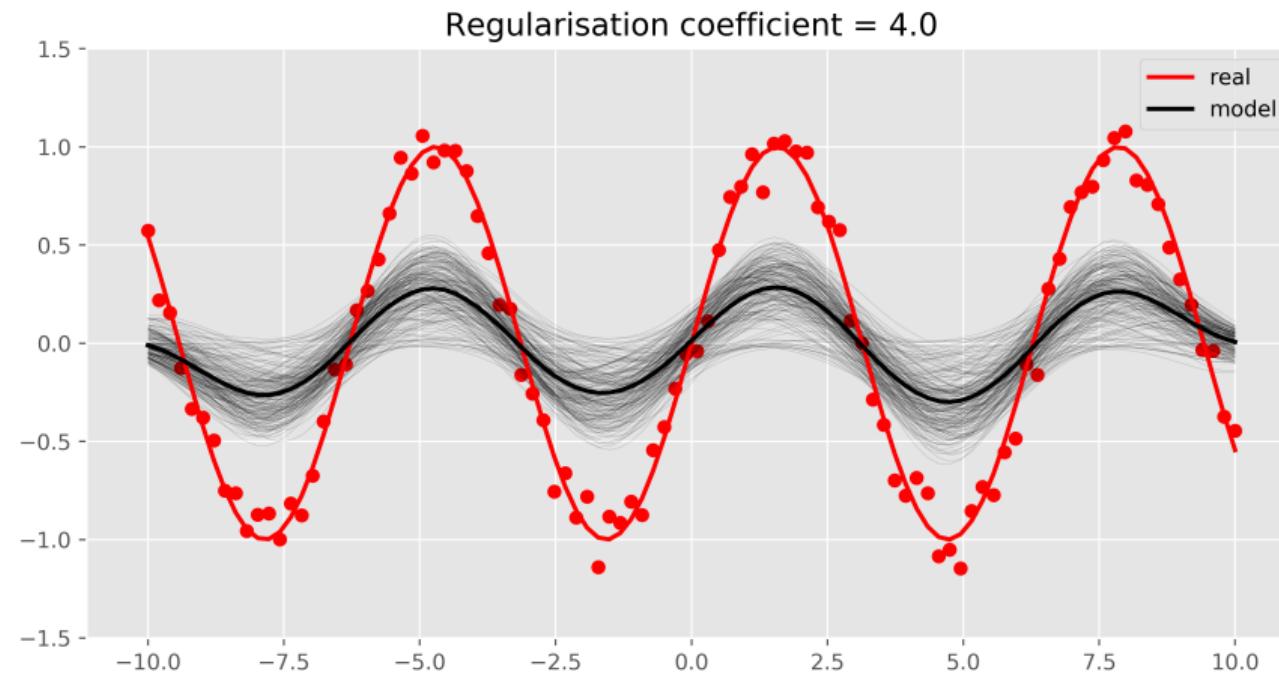
Regularised Linear Regression

Medium Variance - Medium Bias...

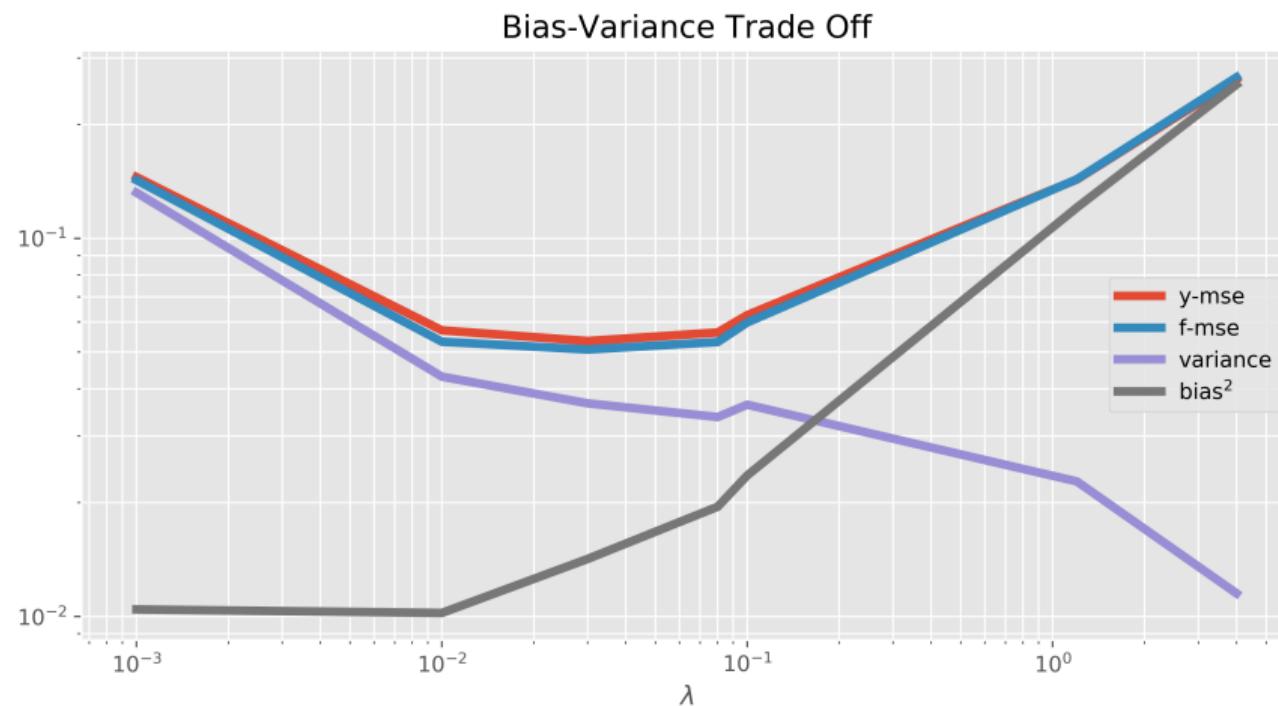


Regularised Linear Regression

Low Variance - High Bias...



Regularised Linear Regression



Choosing λ ...

- In reality we do not have access to the true function f^* and so evaluating $\text{Bias}[\tilde{f}]$ is impossible
- We *can* estimate the expected mean squared error over a test set held out from training and use this to tune λ (cross validation)
- But relies on splitting our dataset reducing the size of N and increasing our chances of overfitting
- Is there a better way?

Overview

1. Linear Regression

2. Frequentist Linear Regression

Linear Regression - A frequentist approach

As optimal function estimation

Learning from noisy observations

Overfitting and problems with finite datasets

Regularisation

3. Bayesian Linear Regression

Our example revisited

The Effective number of Parameters

Bayesian Linear Regression

Bayesian Linear regression offers a valuable alternative...

1. Assume that the weights are no longer fixed but instead a *random variable* $\mathbf{w} \sim p(\mathbf{w}|\beta)$ with *prior* $p(\mathbf{w}|\beta) = \text{Norm}(\mathbf{w}|0, \beta^{-1}\mathbf{w})$ (shape of prior governed by the prior *precision* β).
2. As before assume the *likelihood* of our observations is given as $p(\mathbf{y}|\mathbf{w}, \alpha) = \prod_{n=1}^N \text{Norm}(y_n|\mathbf{w}^\top \mathbf{x}_n, \alpha^{-1})$
3. Calculate the posterior distribution $p(\mathbf{w}|\mathbf{y}, \alpha, \beta)$. In this case we have

$$p(\mathbf{w}|\mathbf{y}, \alpha, \beta) = \text{Norm}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \quad \text{with} \quad \mathbf{m}_N = \alpha \mathbf{S}_N \Phi^\top \mathbf{y} \quad \text{and} \quad \mathbf{S}_N^{-1} = \beta \mathbf{I} + \alpha \Phi^\top \Phi$$

4. Use the *posterior-predictive* distribution

$$p(\mathbf{y}|\mathbf{y}, \alpha, \beta) = \int p(\mathbf{y}|\mathbf{w}) p(\mathbf{w}|\mathbf{y}, \alpha, \beta) d\mathbf{w} = \text{Norm}(\mathbf{y}|\mathbf{m}_N^\top \Phi, \alpha^{-1} + \Phi^\top \mathbf{S}_N \Phi)$$

to predict *new y* averaging across *all* possible values of weights \mathbf{w} .

Bayesian Linear Regression

What about the hyper-parameters α and β ?

- Could assume conjugate priors and add them to the latents adopting a full Bayesian approach BUT results in intractable posterior distribution for w
- Instead consider maximum likelihood estimators using $\hat{\alpha}, \hat{\beta} = \arg \max_{\alpha, \beta} p(\mathbf{y} | \alpha, \beta)$ maximising the *marginal likelihood* $p(\mathbf{y} | \alpha, \beta) = \int p(\mathbf{y} | \mathbf{w}, \alpha) p(\mathbf{w} | \beta) d\mathbf{w}$ → Known as *Empirical Bayes* or *Type II Maximum Likelihood*
- Marginal likelihood is in this case given as

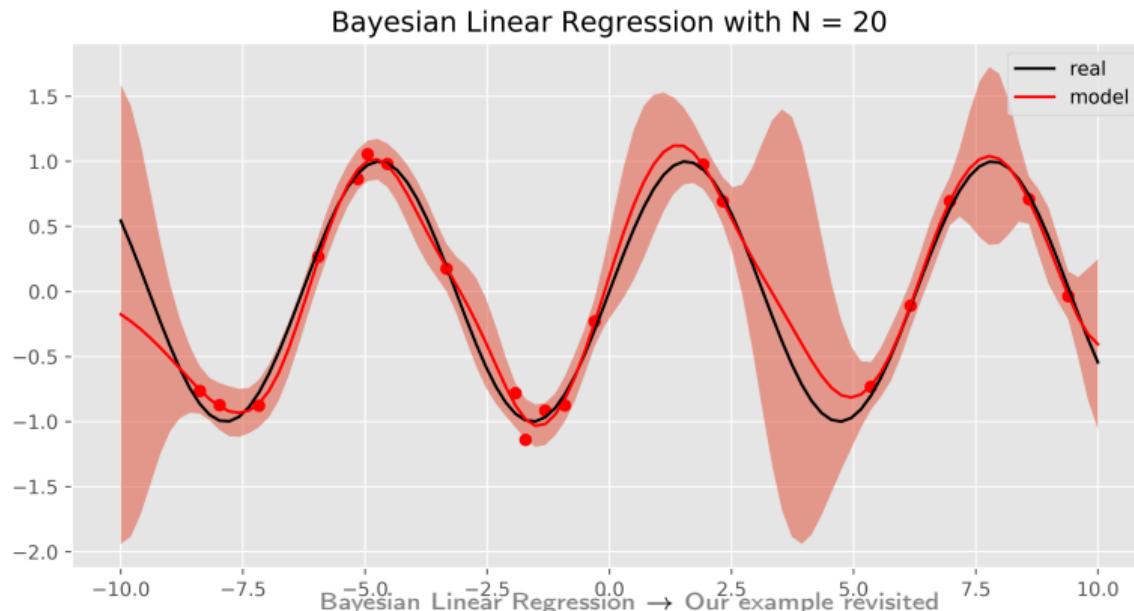
$$\log p(\mathbf{y} | \alpha, \beta) = \frac{M}{2} \log \beta + \frac{N}{2} \log \alpha - \frac{\alpha}{2} \|\mathbf{y} - \Phi \mathbf{m}_N\|^2 - \frac{\beta}{2} \mathbf{m}_N^\top \mathbf{m}_N - \frac{1}{2} |\mathbf{S}_N^{-1}| - \frac{N}{2} \log 2\pi$$

- Differentiating and setting to 0 an iterative scheme emerges for estimating $\hat{\alpha}$ and $\hat{\beta}$...
 1. Choose starting values for α and β
 2. Calculate $\lambda = \text{eig}(\Phi^\top \Phi)$
 3. Calculate $\mathbf{S}_N = (\beta I + \alpha \Phi^\top \Phi)^{-1}$
 4. Calculate $\mathbf{m}_N = \alpha \mathbf{S}_N \Phi^\top \mathbf{y}$
 5. Calculate $\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \beta}$
 6. Update $\alpha = \frac{(N - \gamma)}{\sum_n (y_n - \mathbf{m}_N^\top \phi(x_n))^2}$
 7. Update $\beta = \frac{\gamma}{\mathbf{m}_N^\top \mathbf{m}_N}$
 8. Repeat steps 3-7 until convergence

Bayesian Linear Regression

Our earlier example revisited...

- Bayesian linear regression *automatically* results in a regularised solution as a result of *marginalising* out the uncertainty in w
- Hyper-parameters α and β are estimated from the *training set* - all available data used for training



What is γ ?

- Noting that $\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \beta}$ it is equal to the sum of terms $\frac{\lambda_i}{\lambda_i + \beta}$.
- As $\Phi^\top \Phi$ is positive definite matrix we have $\lambda_i > 0$ and so $0 \leq \frac{\lambda_i}{\lambda_i + \beta} \leq M$
- Two cases:
 1. $\lambda_i \gg \beta$: the estimate for w_i will be close to its ML estimate and $\frac{\lambda_i}{\lambda_i + \beta} \approx 1$
 2. $\lambda_i \ll \beta$: The estimate for $w_i \approx 0$ (the prior value) and $\frac{\lambda_i}{\lambda_i + \beta} \approx 0$
- The quantity γ has an intuitive interpretation as the **effective number of parameters** in the model.
- γ parameters are set by ML and $M - \gamma$ are set from the prior.
- This can be seen in the estimate for the likelihood variance $\alpha^{-1} = \frac{1}{N-\gamma} \sum_n (y_n - \mathbf{m}_n^\top \phi(\mathbf{x}_m))^2$