

Lecture 5. Regularisation

COMP90051 Statistical Machine Learning

Semester 2, 2018

Lecturer: Ben Rubinstein



THE UNIVERSITY OF
MELBOURNE

Copyright: University of Melbourne

This lecture

- Regularisation
 - * Irrelevant features and an ill-posed problem
 - * Regulariser as prior
 - * Model complexity
 - * Constrained modelling
 - * Bias-variance trade-off

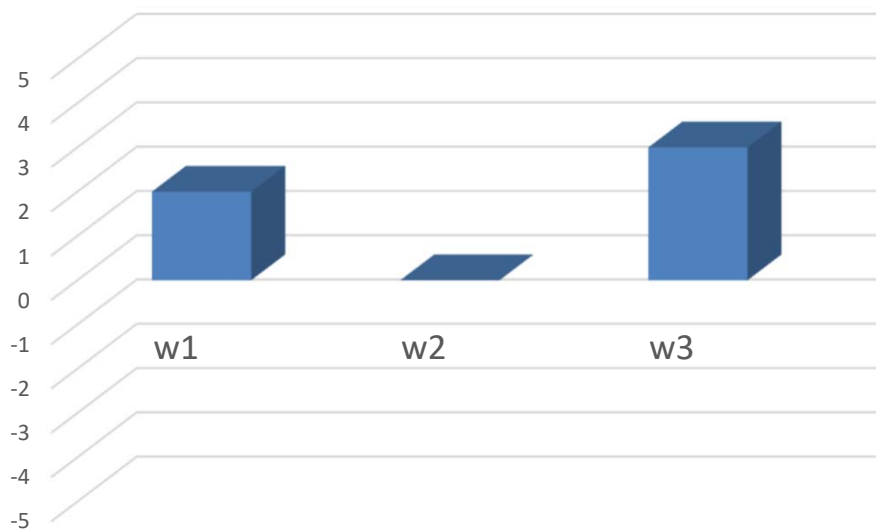
Regularisation

Process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting

- Major technique & theme, throughout ML
- Addresses one or more of the following related problems
 - * Avoids ill-conditioning
 - * Introduce prior knowledge
 - * Constrain modelling
- This is achieved by augmenting the objective function
- In this lecture: we cover the first two aspects. We will cover more of regularisation throughout the subject

Example 1: Feature importance

- Linear model on three features
 - * \mathbf{X} is matrix on $n = 4$ instances (rows)
 - * Model: $y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$



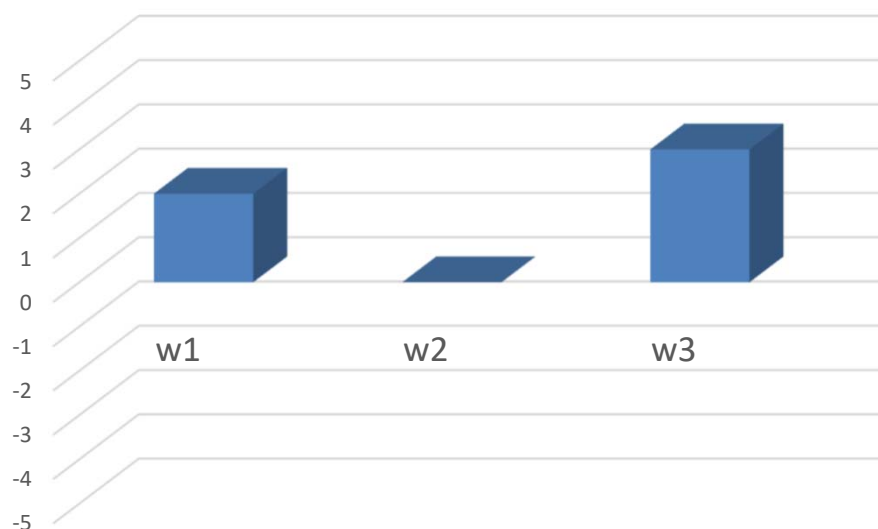
Question: Which feature is more important?

Question: Which feature is more important?

1
2
3
I don't
know

Example 1: Feature importance

- Linear model on three features
 - * \mathbf{X} is matrix on $n = 4$ instances (rows)
 - * Model: $y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$

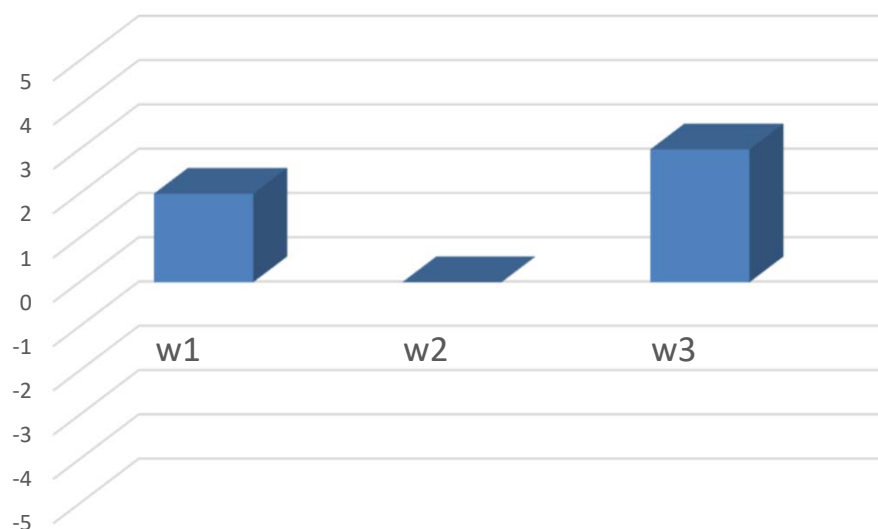


Example 1: Irrelevant features

- Linear model on three features, first two same

- * \mathbf{X} is matrix on $n = 4$ instances (rows)
- * Model: $y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$
- * First two columns of \mathbf{X} identical
- * Feature 2 (or 1) is irrelevant

3	3	7
6	6	9
21	21	79
34	34	2



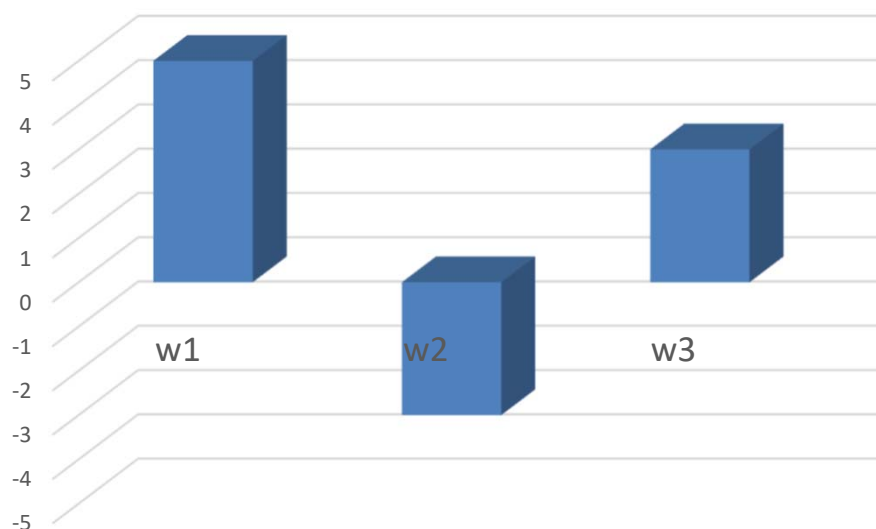
- Effect of perturbations on model predictions?

- * Add Δ to w_1
- * Subtract Δ from w_2

Example 1: Irrelevant features

- Linear model on three features, first two same
 - * \mathbf{X} is matrix on $n = 4$ instances (rows)
 - * Model: $y = w_1x_1 + w_2x_2 + w_3x_3 + w_0$
 - * First two columns of \mathbf{X} identical
 - * Feature 2 (or 1) is **irrelevant**

3	3	7
6	6	9
21	21	79
34	34	2



- Effect of perturbations on model predictions?
 - * Add Δ to w_1
 - * Subtract Δ from w_2

Question: Which feature is more important?

1
2
3
I don't
know

Problems with irrelevant features

- In example, suppose $[\hat{w}_0, \hat{w}_1, \hat{w}_2, \hat{w}_3]'$ is “optimal”
- For any δ new $[\hat{w}_0, \hat{w}_1 + \delta, \hat{w}_2 - \delta, \hat{w}_3]'$ get
 - * *Same* predictions!
 - * *Same* sum of squared errors!
- Problems this highlights
 - * The solution is not **unique**
 - * Lack of **interpretability**
 - * Optimising to learn parameters is **ill-posed problem**

Irrelevant (co-linear) features in general

- Extreme case: features complete clones
- For linear models, more generally
 - * Feature $X_{.j}$ is irrelevant if
 - * $X_{.j}$ is a **linear combination** of other columns

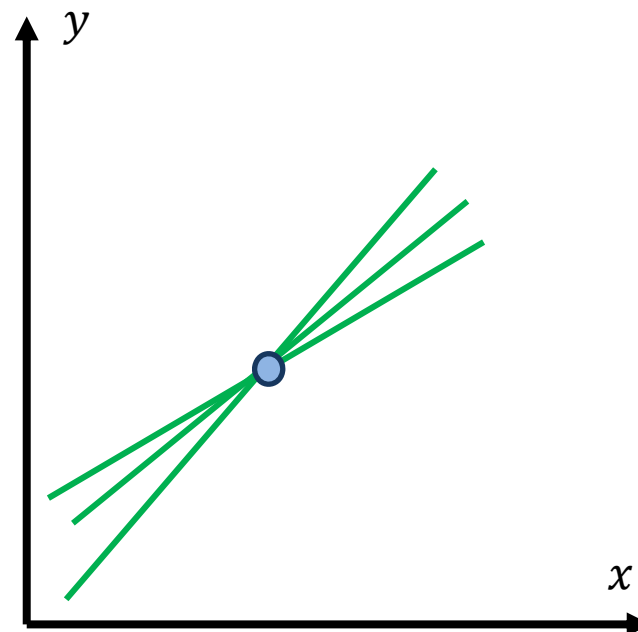
$$X_{.j} = \sum_{l \neq j} \alpha_l X_{.l}$$

... for some scalars α_l

- Even *near*-irrelevance can be problematic
 - * Zero, or v small eigenvalues of $X'X$
- Not just a pathological extreme; *easy to happen!*

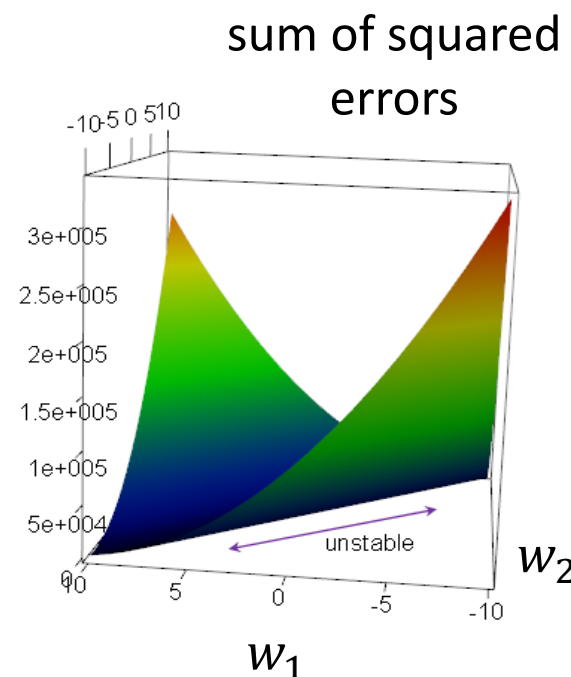
Example 2: Lack of data

- Model is more complex than data
- Extreme example:
 - * Model has two parameters (slope and intercept)
 - * Only one data point
- Underdetermined system



Ill-posed problems

- In both examples, finding the best parameters becomes an **ill-posed problem**
- This means that the problem solution is not defined
 - * In our case w_1 and w_2 cannot be uniquely identified
- Remember the normal equations solution $\hat{\mathbf{w}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$
- With irrelevant features, $\mathbf{X}'\mathbf{X}$ has **no inverse**
- The system of linear equations has more unknowns than equations



convex, but not
strictly convex

Re-conditioning the problem

- Regularisation: introduce an **additional condition** into the system

- The original problem is to minimise $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$

- The regularised problem is to minimise

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \text{ for } \lambda > 0$$

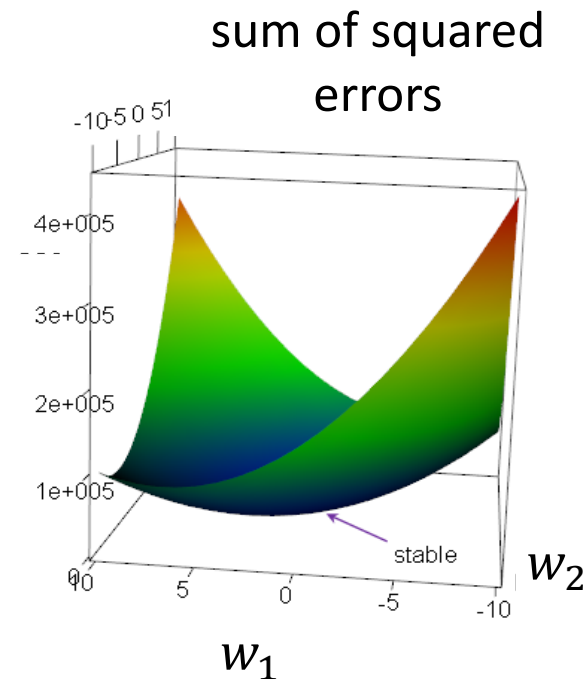
- The solution is now

$$\hat{\mathbf{w}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$



- This formation is called **ridge regression**

- * Turns the ridge into a peak
- * Adds λ to eigenvalues of $\mathbf{X}'\mathbf{X}$: makes invertible



strictly convex

Regulariser as a prior

- Without regularisation, parameters found based entirely on the information contained in the training set \mathbf{X}
 - * Regularisation introduces **additional information**
- Recall our probabilistic model $Y = \mathbf{x}'\mathbf{w} + \varepsilon$
 - * Here Y and ε are random variables, where ε denotes noise
- Now suppose that \mathbf{w} is also a random variable (denoted as \mathbf{W}) with a Normal **prior distribution**

$$\mathbf{W} \sim \mathcal{N}(0, \lambda^2)$$

- * I.e. we expect small weights and that no one feature dominates
- * Is this always appropriate? E.g. data centring and scaling
- * We could encode much more elaborate problem knowledge

Computing posterior using Bayes rule

- The prior is then used to compute the posterior

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

- Instead of maximum likelihood (MLE), take **maximum a posteriori** estimate (MAP)

- Apply log trick, so that
 $\log(\text{posterior}) = \log(\text{likelihood}) + \log(\text{prior}) - \log(\text{marg})$

- Arrive at the problem of minimising

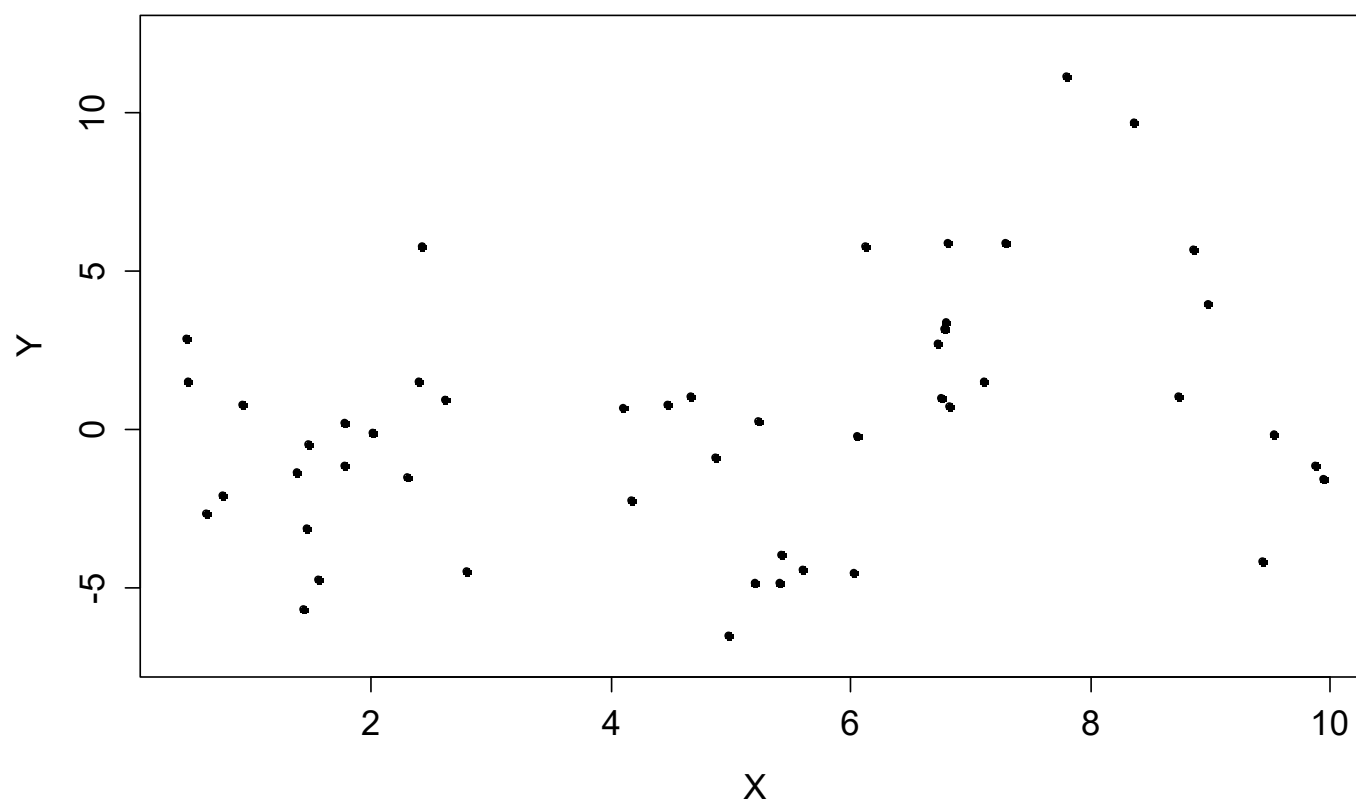
$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

this term doesn't
affect optimisation

Regularisation in Non-Linear Models

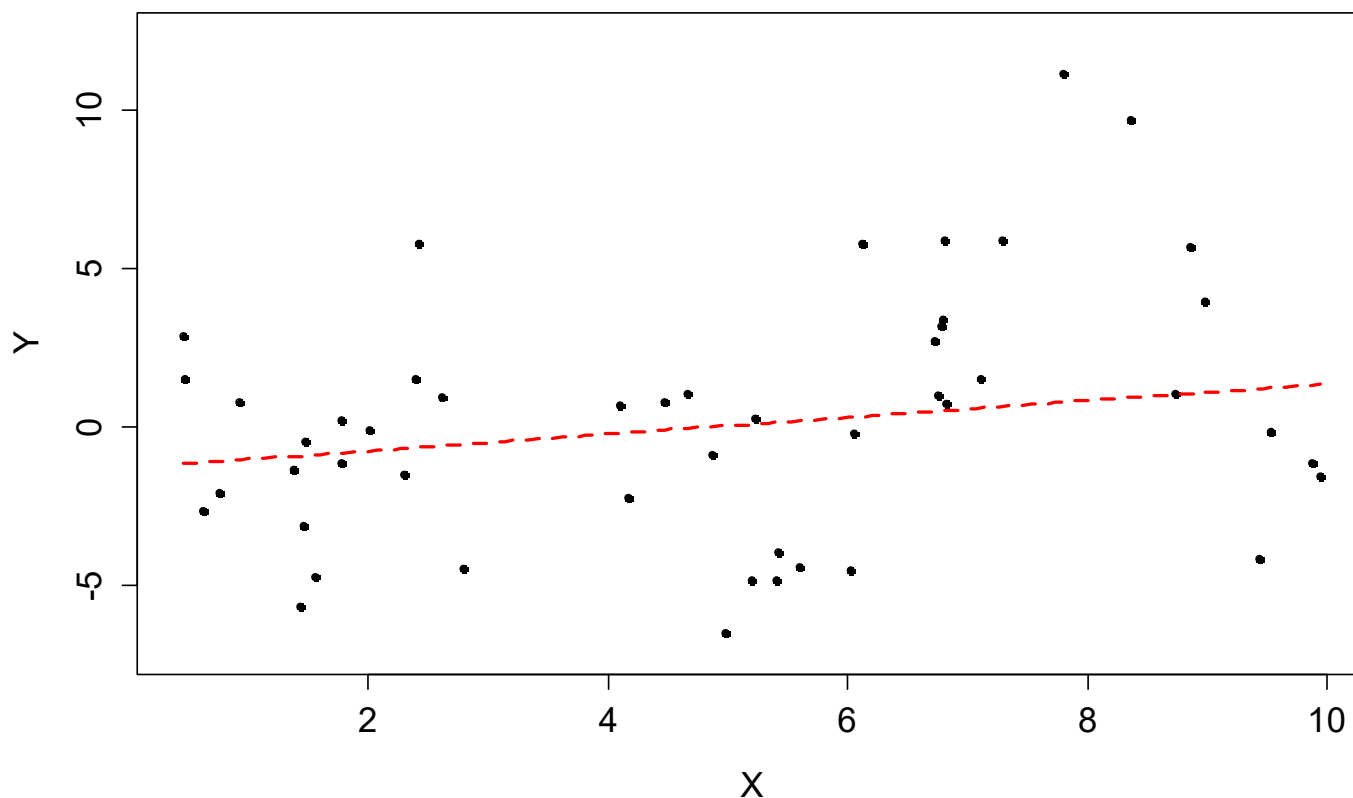
Model selection in ML

Example regression problem



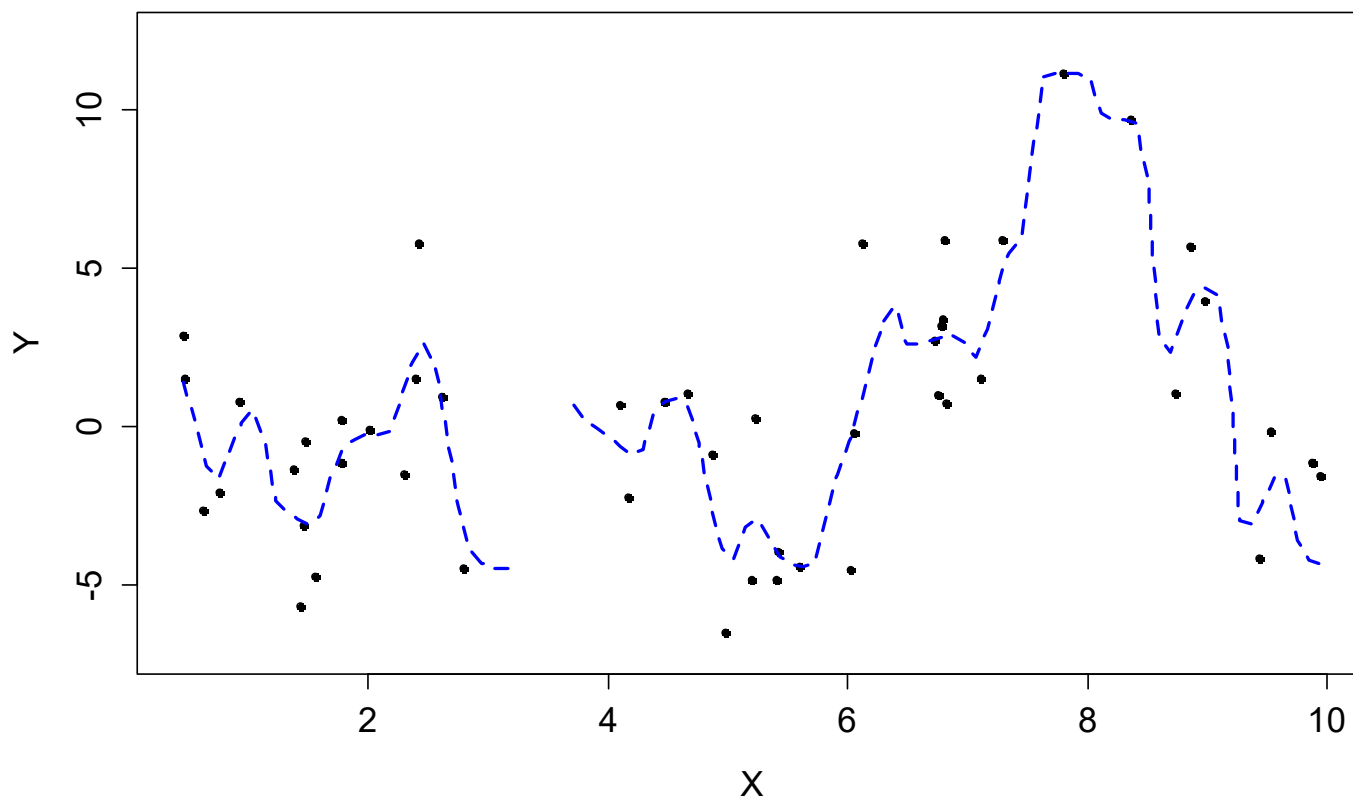
How complex a model should we use?

Underfitting (linear regression)



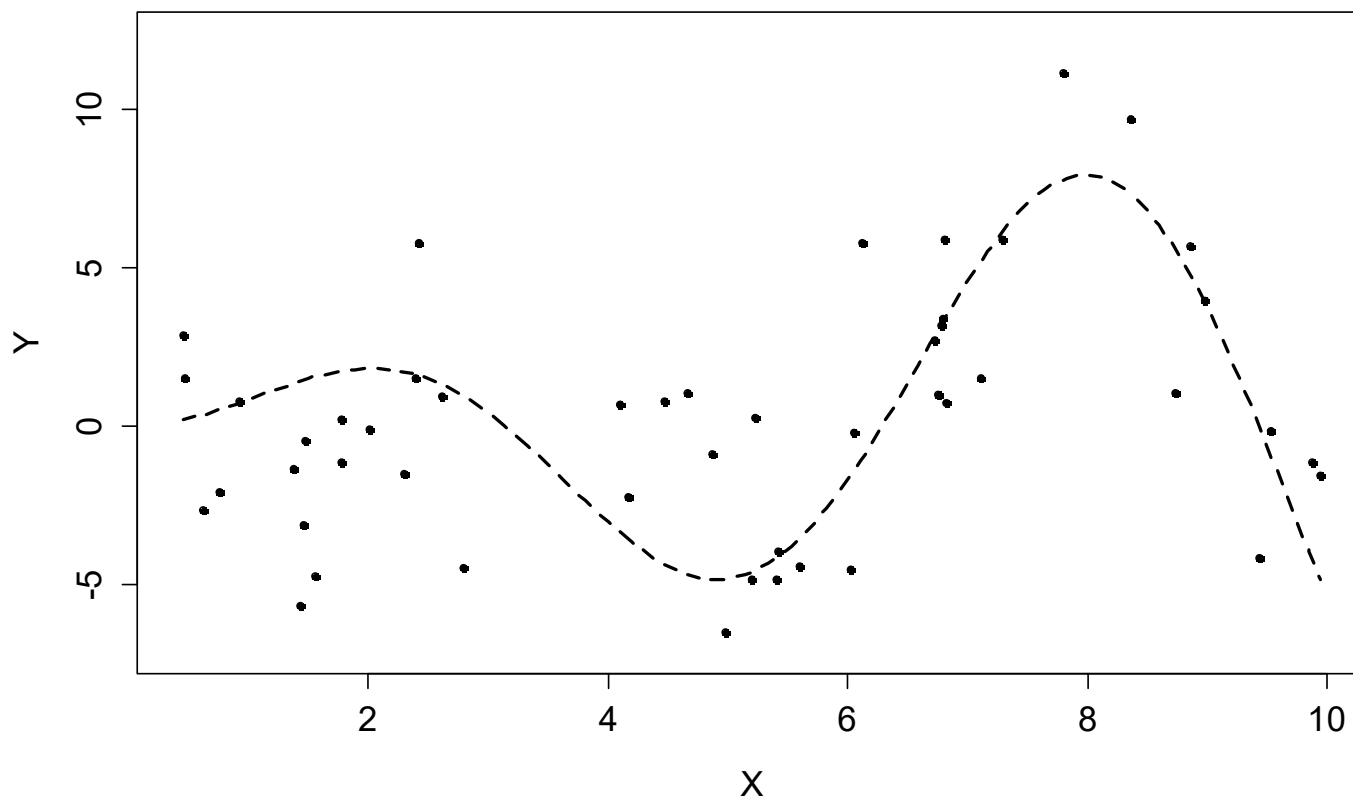
Model class Θ can be **too simple** to possibly fit true model.

Overfitting (non-parametric smoothing)



Model class Θ can be **so complex** it can fit true model + noise

Actual model ($x \sin x$)



The **right model class** Θ will sacrifice some training error, for test error.

How to “vary” model complexity

- Method 1: Explicit model selection
- Method 2: Regularisation
- Usually, method 1 can be viewed a special case of method 2

1. Explicit model selection

- Try different classes of models. Example, try polynomial models of various degree d (linear, quadratic, cubic, ...)
- Use held out validation (cross validation) to select the model
 1. Split training data into D_{train} and $D_{validate}$ sets
 2. For each degree d we have model f_d
 1. Train f_d on D_{train}
 2. Test f_d on $D_{validate}$
 3. Pick degree \hat{d} that gives the best test score
 4. Re-train model $f_{\hat{d}}$ using all data

2. Vary complexity by regularisation

- Augment the problem:

$$\hat{\boldsymbol{\theta}} \in \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} (L(\text{data}, \boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}))$$

- E.g., ridge regression

$$\hat{\mathbf{w}} \in \operatorname{argmin}_{\mathbf{w} \in W} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

- Note that regulariser $R(\boldsymbol{\theta})$ does not depend on data
- Use held out validation/cross validation to choose λ

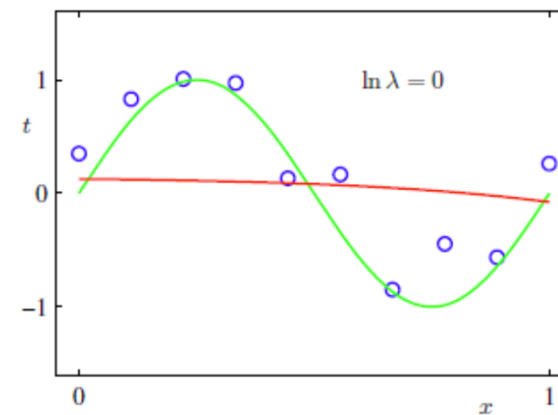
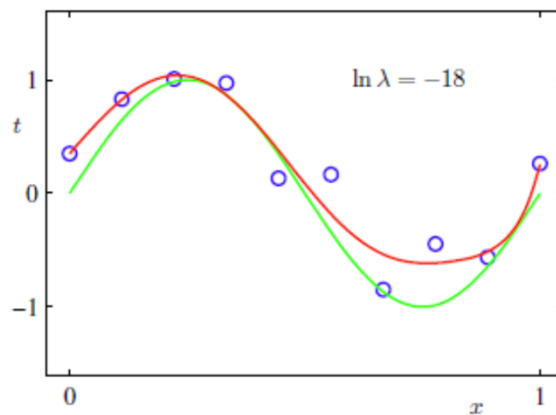
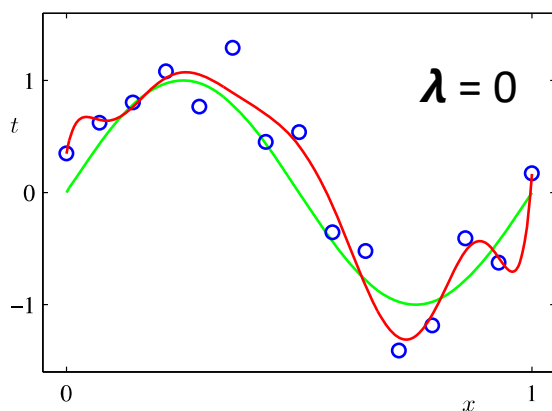
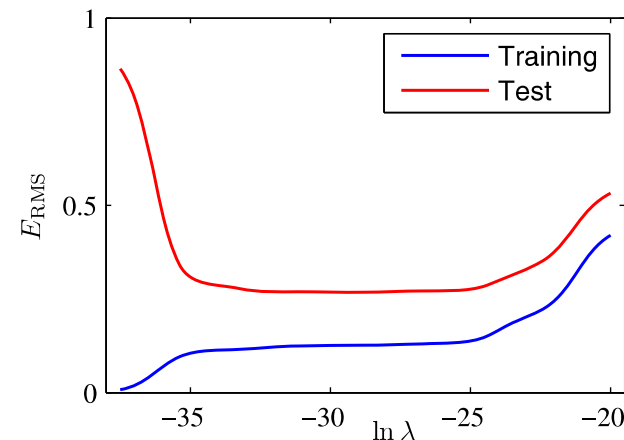
Example: Polynomial regression

- 9th-order polynomial regression

- * model of form

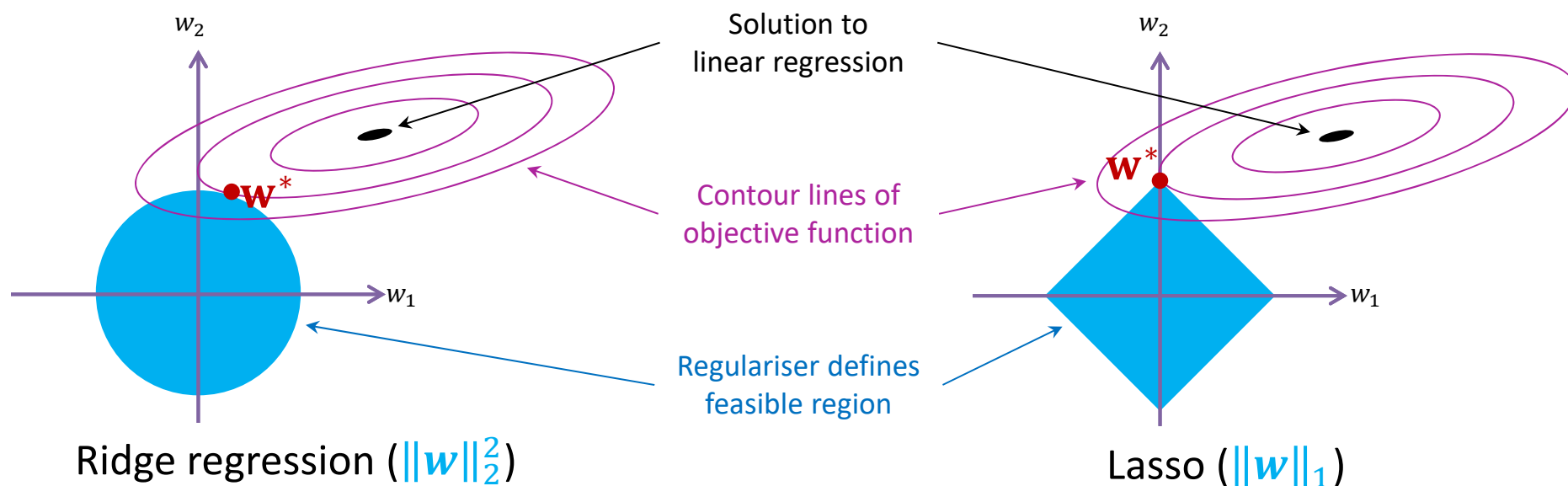
$$\hat{f} = w_0 + w_1 x + \dots + w_9 x^9$$

- * regularised with $\lambda \|\mathbf{w}\|_2^2$ term



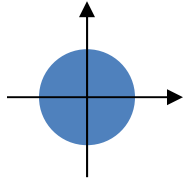
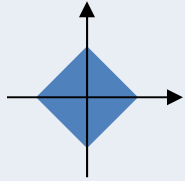
Regulariser as a constraint

- For illustrative purposes, consider a *modified problem*:
minimise $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$ subject to $\|\mathbf{w}\|_2^2 \leq \lambda$ for $\lambda > 0$



- Lasso (L_1 regularisation)** encourages solutions to sit on the axes
 → Some of the weights are set to zero → **Solution is sparse**

Regularised linear regression

Algorithm	Minimises	Regulariser	Solution
Linear regression	$\ \mathbf{y} - \mathbf{X}\mathbf{w}\ _2^2$	None	$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ (if inverse exists)
Ridge regression	$\ \mathbf{y} - \mathbf{X}\mathbf{w}\ _2^2 + \lambda\ \mathbf{w}\ _2^2$	L ₂ norm 	$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$
Lasso	$\ \mathbf{y} - \mathbf{X}\mathbf{w}\ _2^2 + \lambda\ \mathbf{w}\ _1$	L ₁ norm 	No closed-form, but solutions are sparse and suitable for high-dim data

Bias-variance trade-off

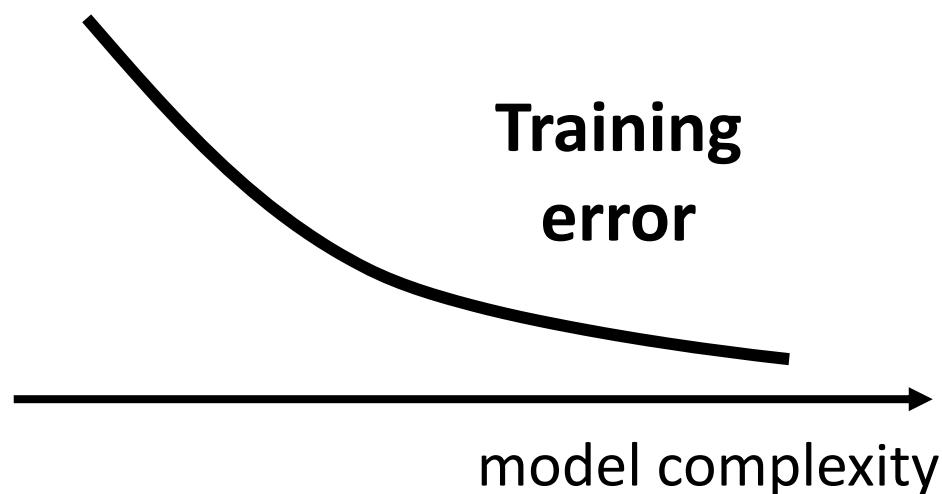
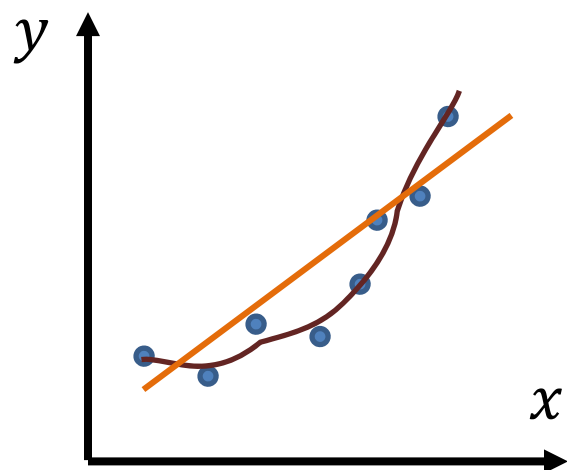
Analysis of relations between
train error, test error and
model complexity

Assessing generalisation capacity

- Supervised learning: train the model on existing data, then make predictions on new data
- Training the model: ERM / minimisation of training error
- Generalisation capacity is captured by risk / test error
- Model complexity is a major factor that influences the ability of the model to generalise
- In this section, our aim is to explore relations between training error, test error and model complexity

Training error and model complexity

- More complex model \rightarrow training error goes down
- Finite number of points \rightarrow usually can reduce training error to 0 (is it always possible?)



(Another) Bias-variance decomposition

- Consider squared loss

$$l(Y, \hat{f}(x_0)) = (Y - \hat{f}(x_0))^2$$

- Lemma: Bias-variance decomposition

$$\mathbb{E} [l(Y, \hat{f}(x_0))] = (\mathbb{E}[Y] - \mathbb{E}[\hat{f}])^2 + \text{Var}[\hat{f}] + \text{Var}[Y]$$

**Risk /
test error
for x_0**

(bias)²

variance

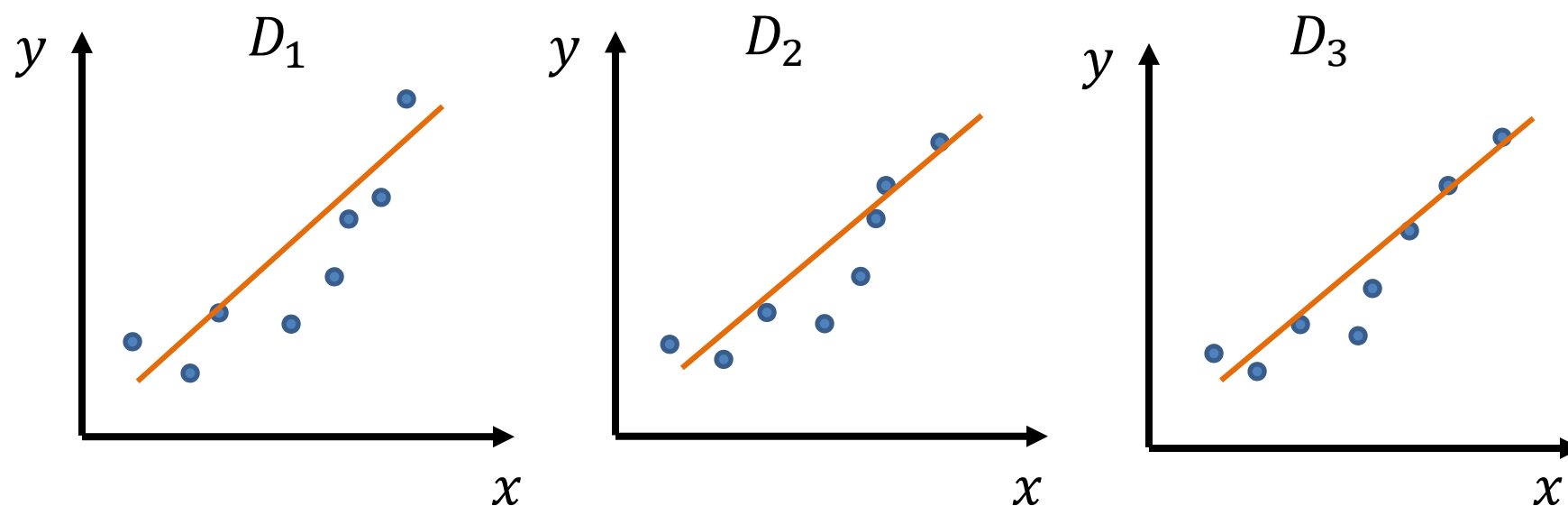
**irreducible
error**

Decomposition proof sketch

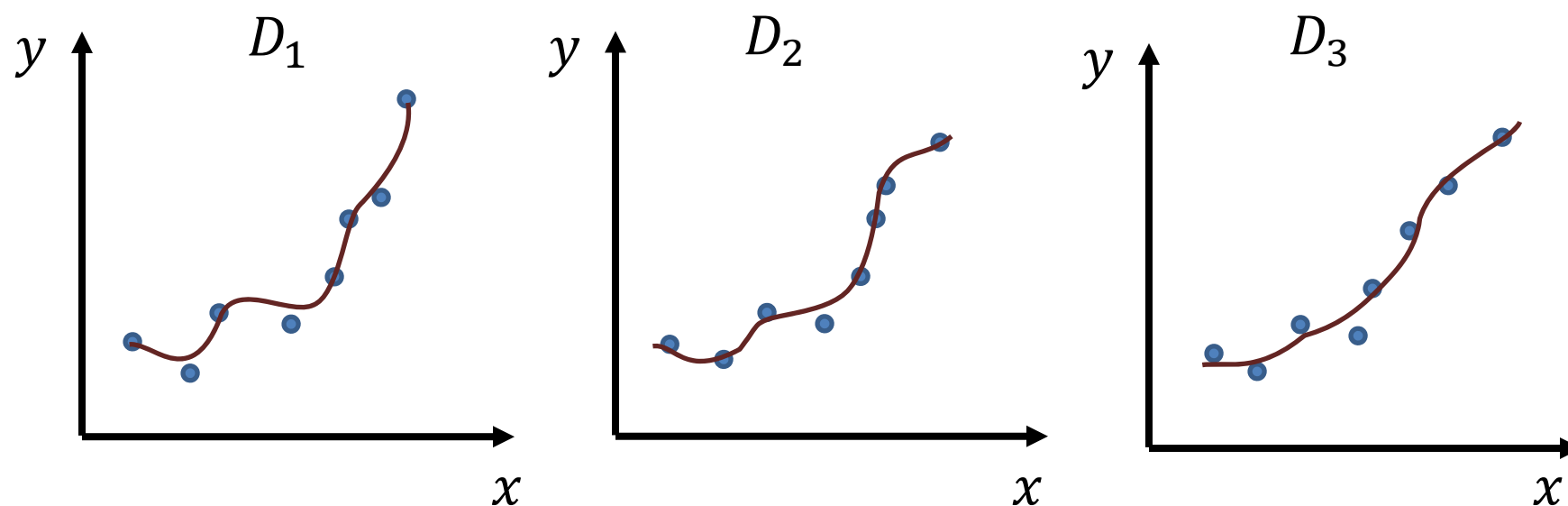
- Here (\mathbf{x}) is omitted to de-clutter notation
- $\mathbb{E} \left[(Y - \hat{f})^2 \right] = \mathbb{E} [Y^2 + \hat{f}^2 - 2Y\hat{f}]$
- $= \mathbb{E} [Y^2] + \mathbb{E} [\hat{f}^2] - \mathbb{E} [2Y\hat{f}]$
- $= \text{Var}[Y] + \mathbb{E}[Y]^2 + \text{Var}[\hat{f}] + \mathbb{E}[\hat{f}]^2 - 2\mathbb{E}[Y]\mathbb{E}[\hat{f}]$
- $= \text{Var}[Y] + \text{Var}[\hat{f}] + \left(\mathbb{E}[Y]^2 - 2\mathbb{E}[Y]\mathbb{E}[\hat{f}] + \mathbb{E}[\hat{f}]^2 \right)$
- $= \text{Var}[Y] + \text{Var}[\hat{f}] + \left(\mathbb{E}[Y] - \mathbb{E}[\hat{f}] \right)^2$

* Green slides are non-examinable

Training data as a random variable

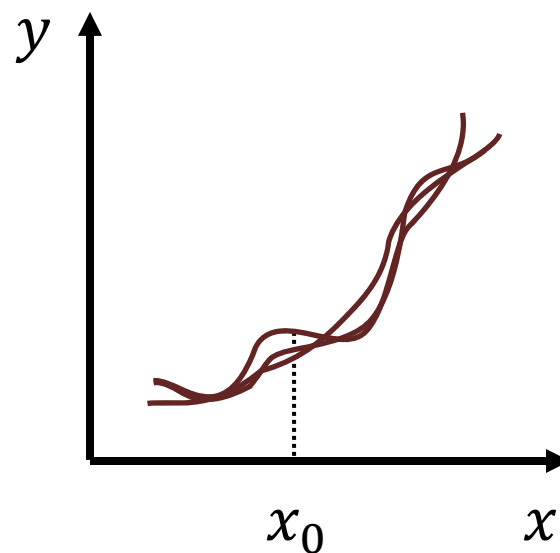
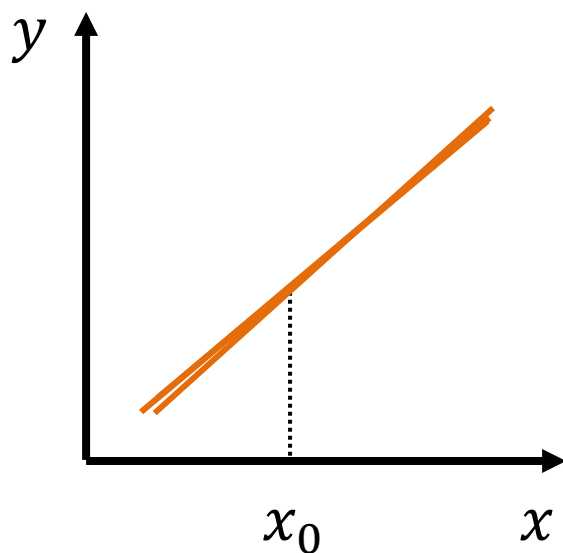


Training data as a random variable



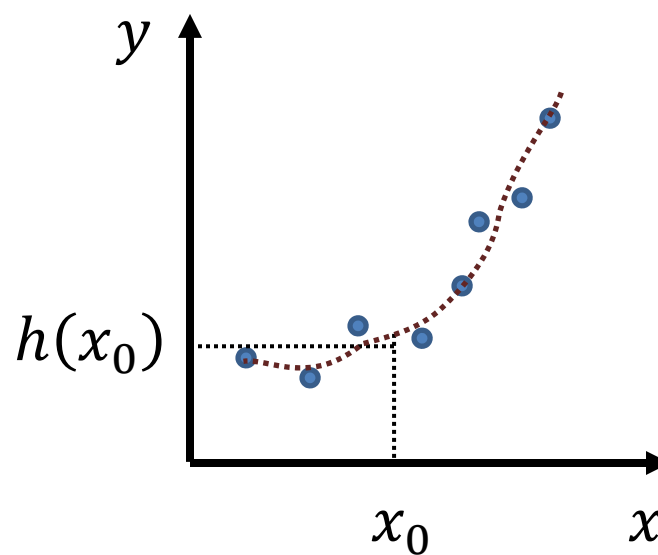
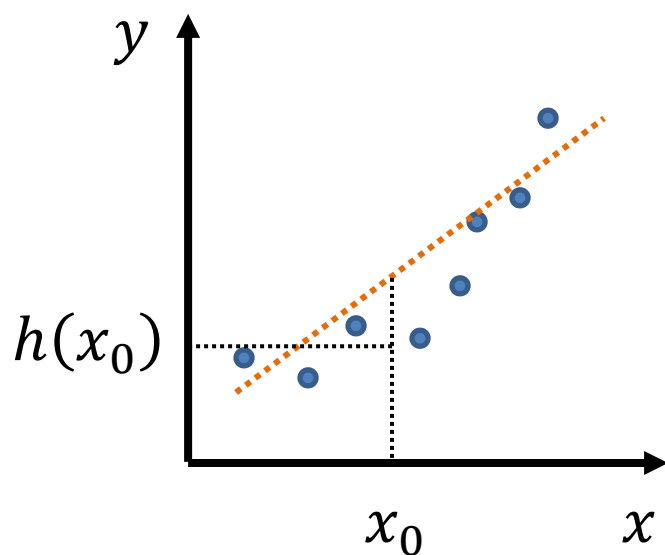
Model complexity and variance

- simple model \rightarrow low variance
- complex model \rightarrow high variance



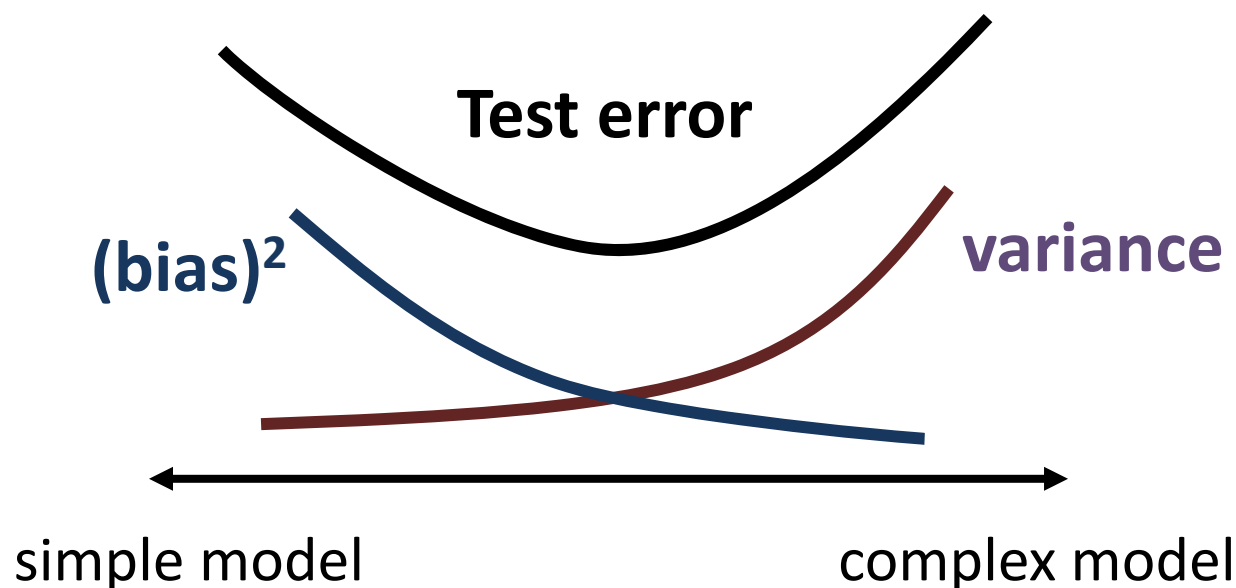
Model complexity and bias

- simple model \rightarrow high bias
- complex model \rightarrow low bias

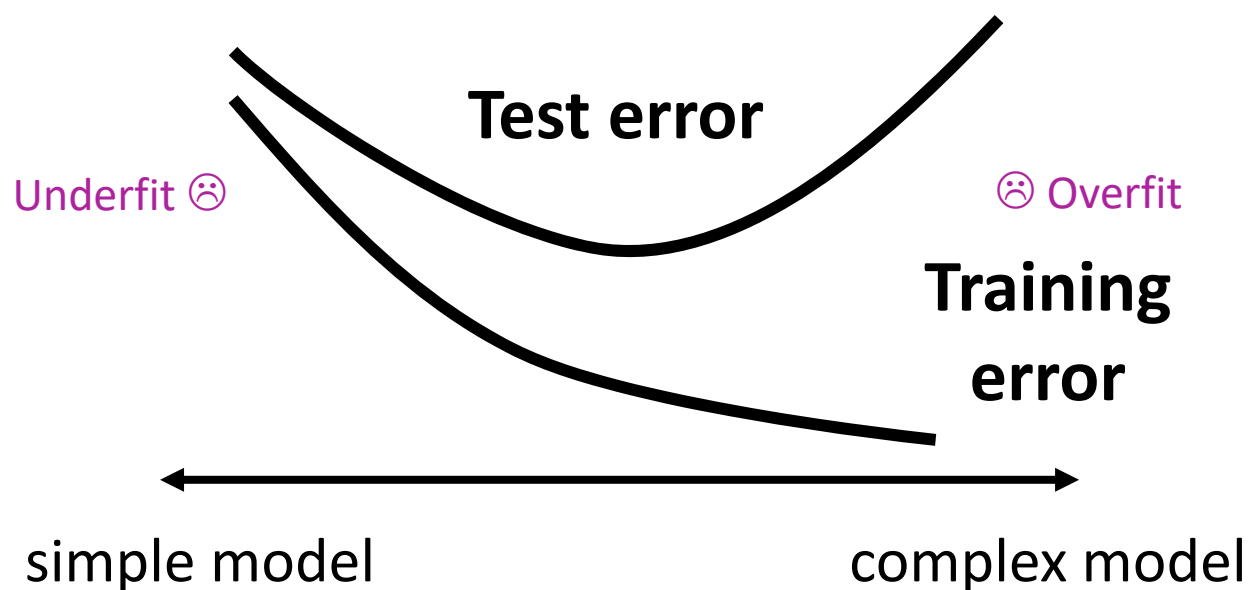


Bias-variance trade-off

- simple model \rightarrow high bias, low variance
- complex model \rightarrow low bias, high variance



Test error and training error



Summary

- Regularisation
 - * Irrelevant features, ill-posed problems
 - * Model complexity
 - * Constrained modelling
 - * Bias-variance trade-off
- **Workshop Week #3**: fun with logistic regression
- Next lecture: Towards neural nets with perceptron