Week 4b:

Model selection

G6061: Fundamentals of Machine Learning [23/24]

Dr. Johanna Senk

Recap: Regularised linear regression

$$\underset{\mathbf{w}}{\text{minimise}} \quad \left\{ \mathcal{L}(y, \hat{f}(x; \mathbf{w})) + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right\}$$

- **Larger** λ, higher regularisation:
  too large, we will not capture any useful trends in the data

- **Smaller** λ, lower regularisation:
  too small, our function will likely be too complex

More regularization tends to cause less overfitting.

UNIVERSITY OF SUSSEX

Outline

At the end of this session, you should be able to:

- Understand what model selection is and why it is an essential part of machine learning
- Understand the decomposition of an error into bias and variance terms, and how model complexity can trade-off between them.
- Be able to explain how validation data can be used for model selection, and to choose regularisation hyper-parameters.

UNIVERSITY OF SUSSEX

Model selection

- Model selection is the process of choosing an appropriate model, in terms of complexity and hyper-parameters,.
- For simple problems, like most we've considered so far, we can choose an appropriate level of complexity just by visual inspection.
- For high-dimensional regression problems, such as predicting variables associated with climate change, it can be less obvious.
- Can you think of a good guiding principle for model selection?

UNIVERSITY OF SUSSEX

Occam's razor

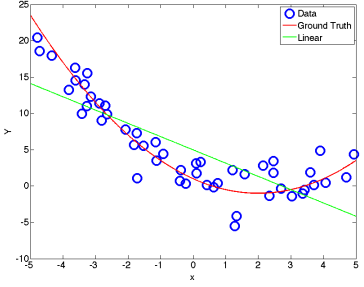
Entities should not be multiplied without necessity.
The simplest explanation is usually the best one.

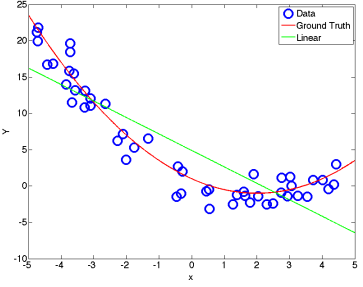
- This is a fundamental principle that is often followed in science, extra complexity needs to be *justifiable*.
- Simple models are easier to test, understand and in the case of ML, fit the parameters.
- Bayesian inference provides a principled solution to reducing model complexity, through regularisation.
- Today we'll talk about methods for interpreting model fitting issues and overcoming them.

UNIVERSITY OF SUSSEX

Model complexity

- The simplest models? Functions that return a constant number or a straight line.
- These models are likely to have a large degree of error!
  - Model too "simple" $\rightarrow$ does not fit the data well

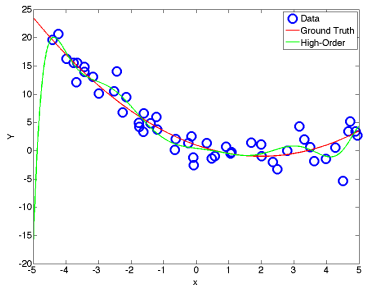


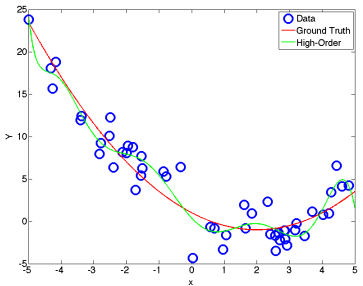


- However, the parameters will be *reliable* to estimate from different subsets of data.
- These models are referred to as *biased*.

Model complexity

- A more complex model will fit the training samples much better.
- However, if the model is too "complex" $\rightarrow$ small changes in the training data lead to large differences in the trained model.



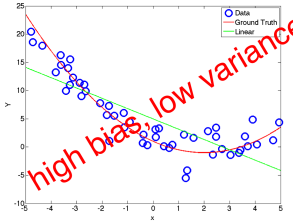


- Because of the variability in model fitting with different training samples, these models are said to have high *variance*.

UNIVERSITY OF SUSSEX

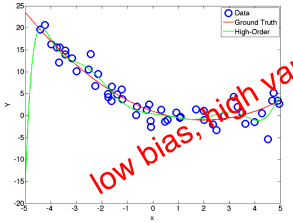
Bias-variance trade-off

- Choice of hypothesis class and hyper-parameters affects bias
  - More complex hypothesis class → less bias
  - More complex hypothesis class → more variance



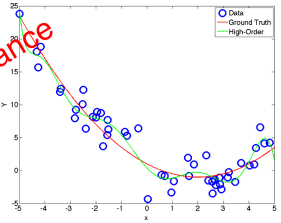
1st degree of polynomial

...



10th degree of polynomial

low bias, low variance
(ideal case)



10th degree of polynomial

UNIVERSITY OF SUSSEX

Bias-variance example

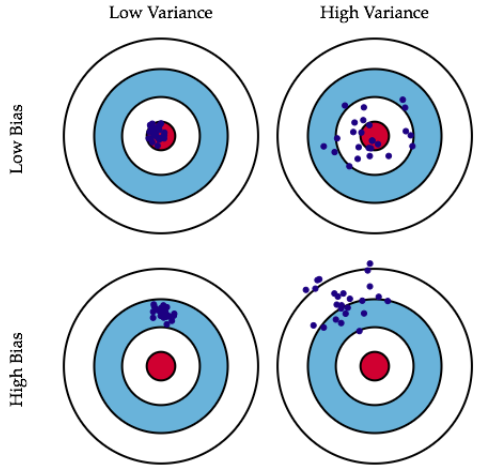


figure is from S. Fortmann-Roe

UNIVERSITY OF SUSSEX

(Squared) bias of predictor

- Given dataset $\mathcal{D}$ with N examples, we would like to learn function $\hat{f}_{\mathcal{D}}(x)$
- Learning a different dataset $\mathcal{D}'$ also with N examples, results in a different $\hat{f}_{\mathcal{D}'}(x)$
- **Expected hypothesis**: $\text{Expectation}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] := \hat{f}_{\text{ave}}(x)$

- **Bias**:
  difference between what you expect to learn $\hat{f}_{\text{ave}}(x)$ and the ground truth $f(x)$
  - Measures how well you expect to represent true solution
  - Decreases with more **complex** model
  - Bias^2 at a single data point x: $(f(x) - \hat{f}_{\text{ave}}(x))^2$
  - Average Bias^2: $\text{Expectation}_x[(f(x) - \hat{f}_{\text{ave}}(x))^2]$

UNIVERSITY OF SUSSEX

Variance of predictor

- Given dataset $\mathcal{D}$ with N examples, we would like to learn function $\hat{f}_{\mathcal{D}}(x)$
- Learning a different dataset $\mathcal{D}'$ also with N examples, results in a different $\hat{f}_{\mathcal{D}'}(x)$
- **Expected hypothesis**: $\text{Expectation}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] := \hat{f}_{\text{ave}}(x)$

- **Variance**:
  difference between what you expect to learn $\hat{f}_{\text{ave}}(x)$ and what you learn from a particular dataset $\hat{f}_{\mathcal{D}}(x)$
  - Measures how sensitive predictor is to specific dataset
  - Decreases with **simpler** model
  - Variance at a single data point x: $\text{Expectation}_{\mathcal{D}}[(\hat{f}_{\mathcal{D}}(x) - \hat{f}_{\text{ave}}(x))^2]$
    Note: $\text{Var}(x) = \text{Expectation}_x[(x - \mu)^2]$
  - Average Variance: $\text{Expectation}_x[\text{Expectation}_{\mathcal{D}}[(\hat{f}_{\mathcal{D}}(x) - \hat{f}_{\text{ave}}(x))^2]]$

UNIVERSITY OF SUSSEX

Bias-variance decomposition of squared error

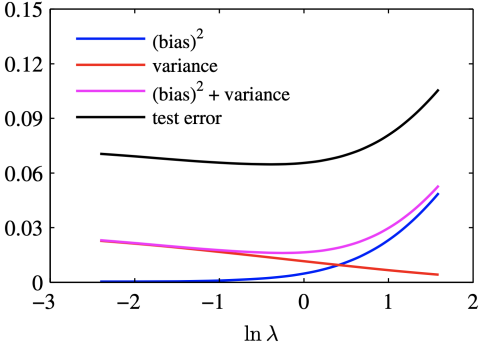
$$\begin{aligned}&\text{Expectation}_{\mathcal{D}}[(\hat{f}_{\mathcal{D}}(x) - f(x))^2] \\ &= \underbrace{\text{Expectation}_{\mathcal{D}}[(\hat{f}_{\mathcal{D}}(x) - \hat{f}_{\text{ave}}(x))^2]}_{\text{variance}(x)} + \underbrace{(\hat{f}_{\text{ave}}(x) - f(x))^2}_{\text{bias}^2(x)}\end{aligned}$$

- **Bias**:
  difference between what you expect to learn $\hat{f}_{\text{ave}}(x)$ and the ground truth $f(x)$
  - More complex hypothesis class $\rightarrow$ less bias
- **Variance**:
  difference between what you expect to learn $\hat{f}_{\text{ave}}(x)$ and what you learn from a particular dataset $\hat{f}_{\mathcal{D}}(x)$
  - More complex hypothesis class $\rightarrow$ more variance

UNIVERSITY OF SUSSEX

Bias-variance decomposition - demonstration

- This approach of understanding model error gives us some insight into the appropriateness of our model complexity.
- For example, you wanted some more intuition into the performance of a regression model, rather than just looking at the squared error.

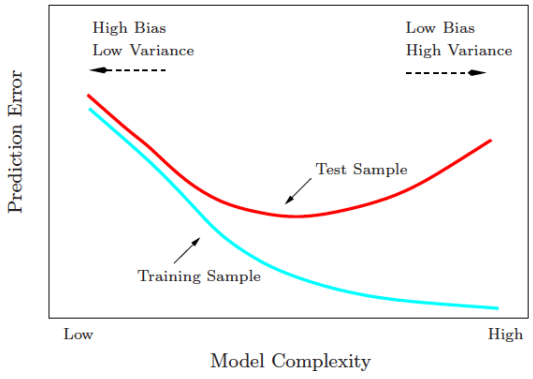


[1] from Bishop, Pattern Recognition and Machine Learning

UNIVERSITY OF SUSSEX

Training and test error as a function of model complexity

For example, the higher the degree of a polynomial, the more complex.



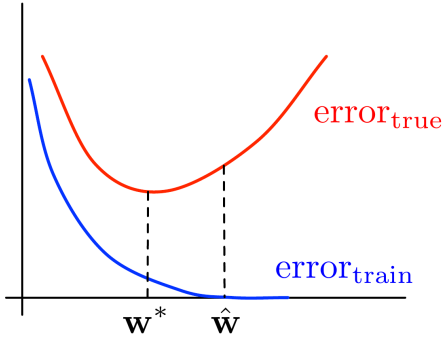
The Elements of Statistical Learning

Extreme case of bias vs. variance

- **Over-fitting**: a learning algorithm overfits the training data if it outputs a solution $\hat{\mathbf{w}}$ when there exists another solution $\mathbf{w}^*$ such that

$$\text{error}_{\text{train}}(\hat{\mathbf{w}}) < \text{error}_{\text{train}}(\mathbf{w}^*) \wedge \text{error}_{\text{true}}(\mathbf{w}^*) < \text{error}_{\text{true}}(\hat{\mathbf{w}})$$

where $\text{error}_{\text{true}}$ is the error at test set and $\text{error}_{\text{train}}$ is the error at training set.



- Low (near zero) bias but very high variance is over-fitting

Analysing machine learning models

- Imagine you're training a model, but it's not going well.
- Common approach: try improving the algorithm in different ways:
  - Try a smaller set of features
  - Try a larger set of features
  - Use a different value for regularisation parameter
  - Try using different machine learning models: naïve Bayes, logistic regression, decision tree, k-Nearest Neighbour, linear perceptron, random forest, etc.
- The approach above might work, but it is very time consuming, and largely a matter of luck whether you end up fixing what the problem really is.

UNIVERSITY OF SUSSEX

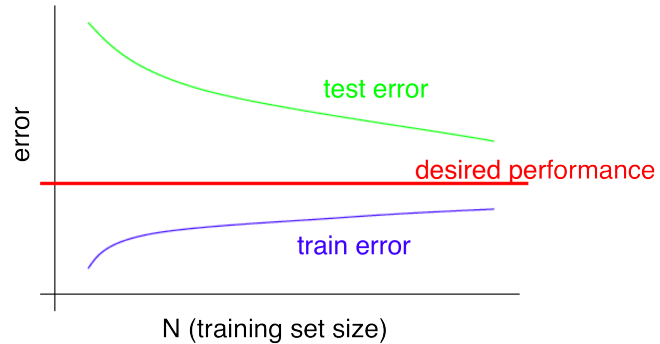
Diagnostic for bias vs. variance

- Better approach:
  - Run diagnostics to figure out what the problem is
  - Fix whatever the problem is
- Suppose you suspect the problem is either:
  - Over-fitting (high variance)
  - Too few features to differentiate positive class from negative class (high bias)
- Diagnostic:
  - High variance: training error will be much lower than test error
  - High bias: training error will also be high

UNIVERSITY OF SUSSEX

More on bias vs. variance

Typical learning curve for **high variance** (at fixed model complexity):

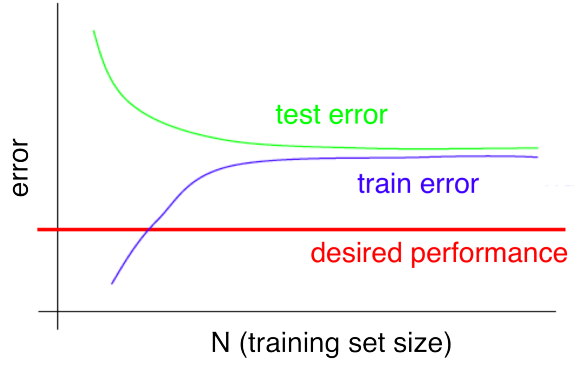


- Validation error still decreasing as N increases.
- Large gap between training and validation error.

figure is from Andrew Ng

More on bias vs. variance

Typical learning curve for **high bias** (at fixed model complexity):



- Even training error is unacceptably high
- Small gap between training and validation error

figure is from Andrew Ng

Diagnostics tell you how to proceed

- Fixes to try:
  - Try a smaller set of features (feature selection) or introduce more regularisation

    Fixes high variance
  - Try a larger set of features (non-linear mapping on features / kernel methods) or reduce regularisation

    Fixes high bias

UNIVERSITY OF SUSSEX

How to choose our hyper-parameters?

- How do we pick the regularisation constant λ
  - and all other constants or parameters in machine learning models: one thing machine learning does not lack is constants to tune!

Training/validation split

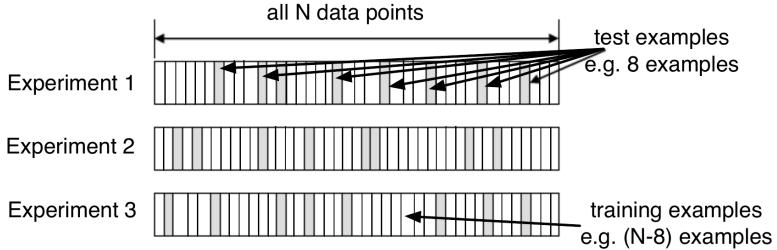
- What we really care about is whether the classifier has learned to *generalise*.
- This can be evaluated by assessing the classification accuracy on a **validation set**.
  - This is a set of labelled data that *was not* used during training.
  - It is assumed that this data is randomly chosen from a set of images that share common characteristics.
  - this is often referred to as "independently and identically distributed" or iid.
  - If the validation set is unusual in some way, it will give us a poor measure of how good our classifier is.
- Penalises the model *overfitting*, i.e., just understanding the training set really well.

UNIVERSITY OF SUSSEX

Random subsampling

Random subsampling performs K data splits of the entire dataset

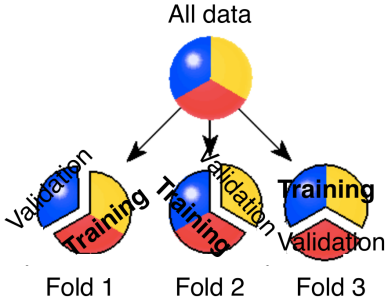
- Each data split randomly selects a fixed number of examples without replacement as test examples
- For each data split we retrain the classifier from scratch with the training examples and then estimate error rate for split i, e_i, with the test examples



- The true error estimate is obtained as the average of the separate estimates e_i

$$e = \frac{1}{K}\sum_{i=1}^{K} e_i$$

K-fold cross-validation



- The dataset is split into K sections, in this case K=3.
- In each run, one fold of data instances is removed from the training set and used to validate or test the model.
- Expected accuracy calculate by averaging over splits:

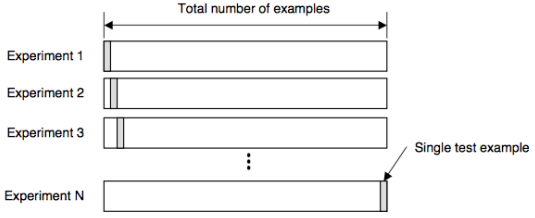
$$e = \frac{1}{K} \sum_{i=1}^{K} e_i$$

UNIVERSITY OF SUSSEX

Leave-One-Out (LOO) cross-validation

Leave-one-out is the degenerate case of K-fold cross-validation, where K is chosen as the total number of examples

- For a dataset with N examples, perform N experiments
- For each experiment use N-1 examples for training and the remaining example for testing



- As before, the true error is estimated as the average error rate on test examples

$$e = \frac{1}{N}\sum_{i=1}^{N} e_i$$

UNIVERSITY OF SUSSEX

Peeking and maintaining a test set

- Having validation sets is all well and good, but it still leaves a problem: as we may make choices based on validation set performance.
- For this reason we might want to keep a separate test set, to evaluate our final performance.
- We *never* look at the test set, until right at the end.
- This is useful if we build a real-life system and need to say how accurate we think it will be.
- If we ever mix our training/validation/testing datasets, this is called peeking. It results in over-inflating our ideas of how well our model will perform.
- Always choose your train/test/split randomly, otherwise you might introduce some odd differences, e.g., the first half of the dataset might only contain cats.

UNIVERSITY OF SUSSEX

Case study

- Model: linear classifier?

$$\mathcal{L}' = \mathcal{L}_{\text{mse}} + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w}$$

- We search λ in the λ-parameter space over $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
- We will use 10-fold stratified cross-validation for each λ and compute accuracy
- Accuracy rate$\pm$ STD result: 10-fold cross-validation table for varying the parameter λ

results are example only!

$\lambda = 10^{-4}$	10^{-3}	10^{-2}	10^{-1}	1
63.15 ± 2.7	66.47 ± 1.8	67.79 ± 1.5	67.27 ± 1.9	63.11 ± 2.5

- Based on the cross-validation results, I will choose $\lambda = 10^{-2}$
- Re-train the classifier with $\lambda = 10^{-2}$ using the whole training dataset and predict the labels for test set where the labels are *unknown*.

UNIVERSITY OF SUSSEX

Case study – more generally

- Magic parameters are everywhere in machine learning models, for example,
  - number of trees, minimum number of instances required to split an internal node, choice of impurity measure in random forest
  - choice of kernel function, value of kernel coefficients, and regularisation parameter in support vector machine
  - choice of regularisation parameter in logistic regression
  - choice of parameter k in k nearest neighbour (kNN) classifier
  - ...
- The more parameters to find, the more computational cost to do cross-validation
  - Suppose we want to find out the best number of trees and minimum samples at leaf nodes in random forest

number of trees \ minimum samples	1	5	10	50
500	?	?	?	?
1,000	?	?	?	?
5,000	?	?	?	?
10,000	?	?	?	?

UNIVERSITY OF SUSSEX

Summary and outlook

- Today we've discussed some of the principles behind model selection: Occam's razor, model complexity, bias-variance trade-off
- We've talked about how to diagnose model training issues, and choose your hyper-parameters:
  training/validation split

Next lecture:

- Neural networks I

UNIVERSITY OF SUSSEX