Machine Learning

Model Selection & Assessment

Dariush Hosseini

dariush.hosseini@ucl.ac.uk
Department of Computer Science
University College London

Lecture Overview

Lecture Overview

By the end of this lecture you should:

1. Understand the concept of **Model Selection** in machine learning
2. Be aware of some practical approaches to model selection, including: **Validation techniques**, **Cross Validation techniques**, and **PAC learning**
3. Know the concept of **Model Assessment**

Lecture Overview

Model Selection

- On several occasions we have seen the need to select various **hyperparameters** in order to fully define our learning algorithm (or our **model**). For example:
  - The ridge regression hyperparameter, λ
  - The degree of polynomial in polynomial regression
  - More abstractly: the 'complexity' measure in the bias-variance trade-off
- How we choose the best hyperparameter / learning algorithm / model for a particular problem is the task of **Model Selection**

Model Selection

- Note that model selection is primarily concerned with **ranking** a set of models on some basis, rather than estimating the **performance** of a particular model

- However, the way in which we perform a ranking is intimately connected with performance:

Model Selection

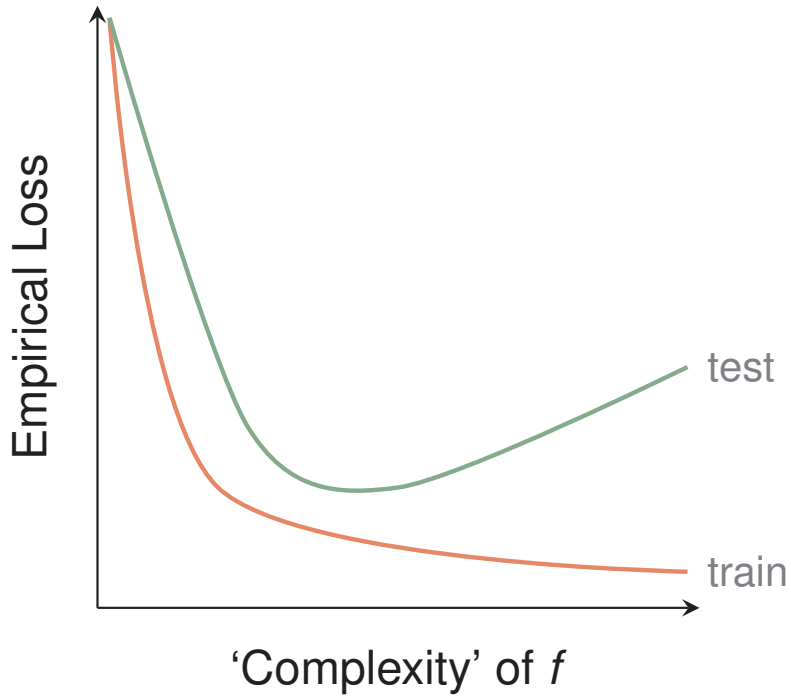
- Often our model selection approaches seek to approximate a measure of performance somehow
- And of course the measure of performance that we are ultimately interested in is the **Generalisation Loss**:

$$\mathbb{E}_{\mathcal{D}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$$

- **Empirical Test Set Loss**, $\mathbb{E}_{\mathcal{S}_{\text{Test}}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$, can act as a reasonable estimator of this quantity...
- ...But cannot be used for model selection!
- Can we use **Empirical Training Set Loss**, $\mathbb{E}_{\mathcal{S}_{\text{Train}}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$ as an alternative?

Training Loss

- No!
  Recall our previous observation that a focus on such a **training loss** when selecting hyperparameters can lead us astray:



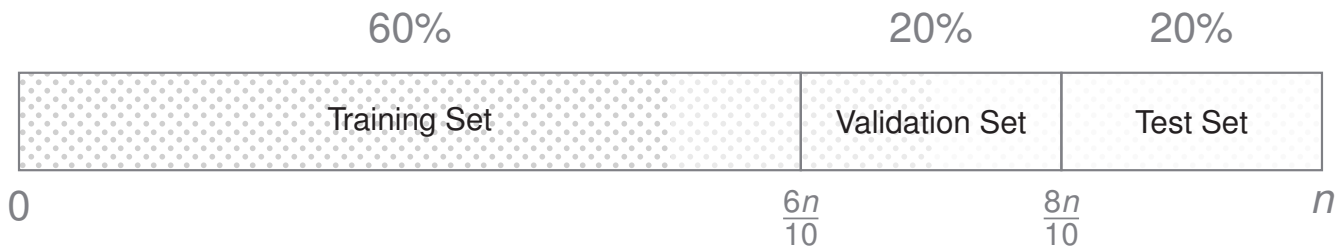
Training Loss

- So **Empirical Risk Minimisation** (ERM) (alone) is not enough to fit our hyperparameters / perform model selection...
- Instead we'll discuss three alternatives:
  - **Validation Techniques**
  - **Cross Validation Techniques**
  - **PAC Learning**

Lecture Overview

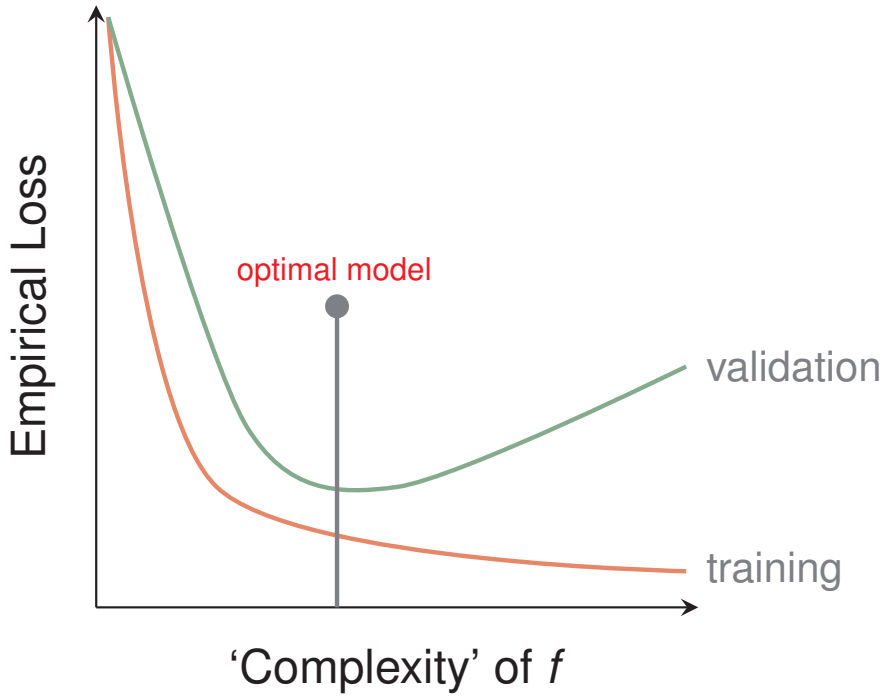
Validation Set Loss

- Here the basic idea is to split the initial training data into three sets: **Training**, **Validation**, and **Test** sets:
  - The first is used for **training** each possible model
  - The second is used to **select** the 'best' model
  - The third is used to **assess** the performance of that model



Validation Set Loss

- The motivation is that the **validation (set) loss** (the empirical loss on the validation set) will provide a proxy for the generalisation loss, which we then use to select an optimal model



Validation Set Bound

- We can prove that the validation loss is related to the generalisation loss, using **Hoeffding's Inequality**
- We will prove the result for the case of misclassification loss, using the following notation:

Notation

- **Validation Set:**

$$\mathcal{S}_{\mathcal{V}} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n_{\mathcal{V}}} \quad \text{where:} \quad \mathcal{S}_{\mathcal{V}} \sim \mathcal{D}^{n_{\mathcal{V}}}$$

- **Loss Measure:**

$$\mathcal{E}(f(\mathbf{x}), y) = \mathbb{I}[y \neq f(\mathbf{x})]$$

- **Generalisation Loss:**

$$\mathsf{L}(\mathcal{E}, \mathcal{D}, f) = \mathbb{E}_{\mathcal{D}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$$

- **Validation Loss:**

$$\mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f) = \frac{1}{n_{\mathcal{V}}} \sum_{i=1}^{n_{\mathcal{V}}} \mathcal{E}(f(\mathbf{x}^{(i)}), y^{(i)})$$

Validation Set Bound

- Let us recall **Hoeffding's Inequality**:
  Let $\mathcal{Z}_1, \ldots, \mathcal{Z}_n$ be independent random variables, such that each $\mathcal{Z}_i$ is bounded by the interval $[a_i, b_i]$, then for any $\epsilon > 0$:

$$\mathbb{P}\left(\mathbb{E}[\overline{\mathcal{Z}}] - \overline{\mathcal{Z}} \geqslant \epsilon\right) \leqslant \exp\left(-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

Validation Set Bound

- Apply this to i.i.d. $\mathcal{Z}_i$, such that $\mathcal{Z}_i = \mathbb{I}[\mathcal{Y}^{(i)} \neq f(\mathcal{X}^{(i)})]$, means that $(b_i - a_i) = 1$ and:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \geqslant \frac{1}{n_{\mathcal{V}}} \sum_{i=1}^{n_{\mathcal{V}}} \mathbb{I}[y^{(i)} \neq f(\mathbf{x}^{(i)})] + \epsilon\right) \leqslant e^{-2n_{\mathcal{V}}\epsilon^2}$$

- And so:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \geqslant \mathbb{E}_{\mathcal{S}_{\mathcal{V}}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + \epsilon\right) \leqslant e^{-2n_{\mathcal{V}}\epsilon^2}$$

Validation Set Bound

- If we wish the probability of the occurence of this event to be no greater than some quantity $\delta \in [0, 1]$ then:

$$\delta > e^{-2n_{\mathcal{V}}\epsilon^2}$$

$$\implies \epsilon < \sqrt{\frac{\ln(1/\delta)}{2n_{\mathcal{V}}}}$$

- This setting of ϵ will ensure that the probability of the occurence of the complement of this event will be greater than $(1 - \delta)$:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] < \mathbb{E}_{\mathcal{S}_{\mathcal{V}}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + \sqrt{\frac{\ln(1/\delta)}{2n_{\mathcal{V}}}}\right) > (1 - \delta)$$

Validation Set Bound

- Equivalently, we can state the **Validation Set Bound**: Let f be some predictor function, then with probability of at least $(1 - \delta)$ over the choice of validation set, $\mathcal{S}_{\mathcal{V}}$:

$$\mathsf{L}(\mathcal{E}, \mathcal{D}, f) \leqslant \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f) + \sqrt{\frac{\ln(1/\delta)}{2n_{\mathcal{V}}}}$$

Validation Set Bound

- This holds for a single function f, but we wish to use the validation loss to choose from amongst a set of functions

- Using the **Union Bound** we can prove that a similar bound holds for all members of a finite hypothesis class, $\mathcal{F}$:

$$\mathcal{F} = \{f_1, ..., f_r\} \qquad \text{of size:} \qquad |\mathcal{F}|$$

Validation Set Bound

- Recall that by the Union Bound, for events A and B, the following inequality holds:

$$\begin{aligned} & \mathbb{P}(\mathsf{A} \vee \mathsf{B}) \leqslant \mathbb{P}(\mathsf{A}) + \mathbb{P}(\mathsf{B}) \\ \implies & 1 - \mathbb{P}(\mathsf{A} \vee \mathsf{B}) \geqslant 1 - (\mathbb{P}(\mathsf{A}) + \mathbb{P}(\mathsf{B})) \\ \implies & \mathbb{P}(\neg(\mathsf{A} \vee \mathsf{B})) \geqslant 1 - (\mathbb{P}(\mathsf{A}) + \mathbb{P}(\mathsf{B})) \\ \implies & \mathbb{P}(\neg\mathsf{A} \wedge \neg\mathsf{B}) \geqslant 1 - (\mathbb{P}(\mathsf{A}) + \mathbb{P}(\mathsf{B})) \end{aligned}$$

by DeMorgan

Validation Set Bound

- Let A be the event, (for some function f^{A}):

$$\mathsf{L}(\mathcal{E}, \mathcal{D}, f^{\mathsf{A}}) > \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f^{\mathsf{A}}) + \sqrt{\frac{\ln\left(1/\delta'\right)}{2n_{\mathcal{V}}}}$$

- Then by the validation set bound:

$$\mathbb{P}(\mathsf{A}) < \delta'$$

- Similarly let B be the event, (for some function f^{B}):

$$\mathsf{L}(\mathcal{E}, \mathcal{D}, f^{\mathsf{B}}) > \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f^{\mathsf{B}}) + \sqrt{\frac{\ln\left(1/\delta'\right)}{2n_{\mathcal{V}}}}$$

- Then by the validation set bound:

$$\mathbb{P}(\mathsf{B}) < \delta'$$

Validation Set Bound

- Combining these, for all $f \in \{f^{\mathsf{A}}, f^{\mathsf{B}}\}$:

$$\mathbb{P}\left(\mathsf{L}(\mathcal{E}, \mathcal{D}, f) \leqslant \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f) + \sqrt{\frac{\ln\left(1/\delta'\right)}{2n_{\mathcal{V}}}}\right) \geqslant 1 - 2\delta'$$

- Then, if we set $\delta = 2\delta'$ then, for all $f \in \{f^{\mathsf{A}}, f^{\mathsf{B}}\}$:

$$\mathbb{P}\left(\mathsf{L}(\mathcal{E}, \mathcal{D}, f) \leqslant \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f) + \sqrt{\frac{\ln\left(2 \times 1/\delta\right)}{2n_{\mathcal{V}}}}\right) \geqslant 1 - \delta$$

- Our result for all $f \in \{\mathcal{F}\}$ where the number of functions contained in $\mathcal{F}$ is $|\mathcal{F}|$ follows in a similar way.

Validation Set Bound

- With probability of at least $(1 - \delta)$ over the choice of $\mathcal{S}_{\mathcal{V}}$:

$$\forall f \in \mathcal{F} \qquad \mathsf{L}(\mathcal{E}, \mathcal{D}, f) \leqslant \mathsf{L}(\mathcal{E}, \mathcal{S}_{\mathcal{V}}, f) + \sqrt{\frac{\ln\left(|\mathcal{F}|/\delta\right)}{2n_{\mathcal{V}}}}$$

Validation Set Bound

- In other words, the validation loss is a good guide to the generalisation loss provided that:
  - $\mathcal{F}$ is not too large (i.e. we don't try too many models)
  - $n_{\mathcal{V}}$ is not too small
- If either of these conditions is not true, then the validation and generalisation losses will drift apart as we begin to **overfit** the model / hyperparameters
- For example the Kaggle leader board phenomenon:
  - Entrants routinely top the validation leader board, but perform poorly on the final test board

Stratification

- One way of alleviating the small sample size problem is to **stratify** our sampling
- **Stratification** involves sampling the data for each set (training/validation/test) such that each contains the same proportion of data points with particular combinations of features
- This helps to ensure that training, validation, and test sets are all representative of $\mathcal{D}$

Lecture Overview

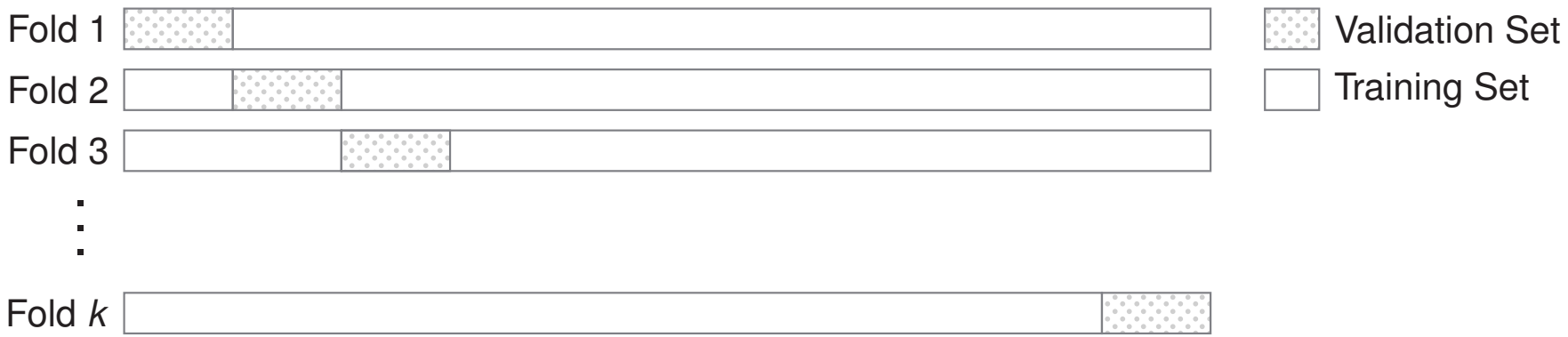
Cross Validation

- But data is often not plentiful at all
- If we wish to retain most of our data for training then we must adopt a different approach:
- **Cross Validation** allows us to perform validation, while still maintaining our training data set

Cross Validation

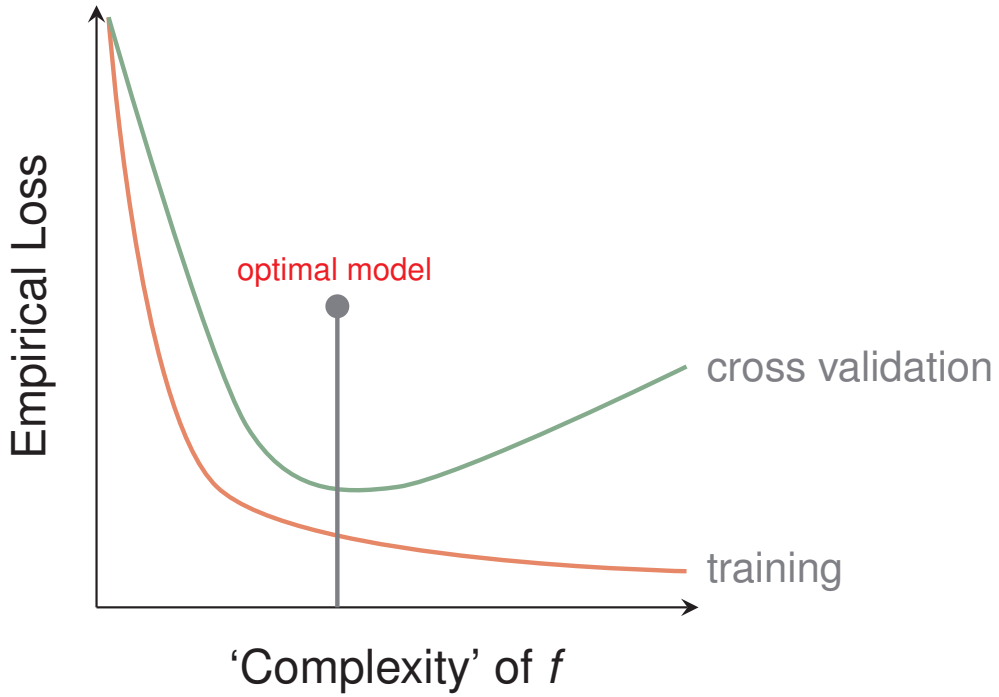
- Here the basic idea is to split the initial training data into k sets (or **folds**)
- The learning algorithm is run k times, for each model, each time using all the folds but one as a **training** set, $\mathcal{S} \setminus \mathcal{S}_k$, and the remaining fold as a **validation** set, $\mathcal{S}_k$
- The validation performance, for a particular model f, is averaged across all k folds to give the **cross validation loss**:

$$\mathsf{L}_{\mathsf{CV}_k}(\mathcal{E}, \mathcal{S}, f) = \frac{1}{k} \sum_{i=1}^{k} \mathsf{L}_{\mathcal{S} \setminus \mathcal{S}_k}(\mathcal{E}, \mathcal{S} \setminus \mathcal{S}_k, f)$$



Cross Validation

- Again, the motivation is that the **cross validation loss** will provide a proxy for the generalisation loss, which we then use to select an optimal model



Cross Validation: k selection

- What value should we select for k?
- $k = n$ is known as **Leave One Out** (**LOO**) cross validation
  - It maximises the training set for each fold iteration - and so reduces bias in the hypothesis learned for each such iteration
  - But it is computationally expensive
  - And each validation fails to give a good estimate of the generalisation error associated with the learned hypothesis
- Empirical analysis suggests that $k = 5$ or 10 is better
  - This is used in practice almost as standard
  - And is computationally cheaper

Cross Validation: Comments

- Cross Validation works well in practice...
- ...But a rigorous understanding of why is an open question
- And in certain circumstances it must be treated with care, for example:
  - **Time Series Data:**
    - Only **sequential** cross validation should be performed
    - Randomising would artificially enhance validation set performance
    - Neighbouring points are likely to be similar and should therefore be kept together

Lecture Overview

The PAC Approach

- Here we begin with the generalisation loss...
- ...And seek to formulate a **probabilistic** 'worst-case' **bound** on this quantity in terms of:
  - The observable **empirical training loss**
  - Some **complexity penalty**, which takes into account the size of the **representation space**, $\mathcal{F}$
- Then we perform model selection by searching for the model / hyperparameter choice which gives rise to the **tightest bound**

PAC Learning

- The learner receives samples and must select a hypothesis, f, from, $\mathcal{F}$, such that with high probability (**probably**) the selected function will have low generalisation loss (will be **approximately correct**)
- This must be true for any probability of success, $(1 - \delta)$, any approximation loss target, ϵ, and any data generating process, $\mathcal{D}$

PAC Learning: Procedure

- These **PAC generalisation bounds** can be generated using the **Uniform Convergence** approach.
  This is a two step procedure:
  - First we seek some finite sample **concentration inequality** that is independent of $\mathcal{D}$. We apply this to one function, f
  - Second we seek to apply this inequality **uniformly** across all f in $\mathcal{F}$ via the **Union Bound** or some measure of the functional complexity of $\mathcal{F}$
- Let's take a look at this approach in a few different **classification** settings:

Notation

- **Training Set:**

$$\mathcal{S} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n} \quad \text{where:} \quad \mathcal{S} \sim \mathcal{D}^n$$

- **Loss Measure:**

$$\mathcal{E}(f(\mathbf{x}), y) = \mathbb{I}[y \neq f(\mathbf{x})]$$

- **Generalisation Loss:**

$$\mathbb{E}_{\mathcal{D}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$$

- **Training Loss:**

$$\mathbb{E}_{\mathcal{S}}\left[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})\right]$$

PAC Bound: Finite $\mathcal{F}$, Consistent Learner

- A **consistent learner** is a learning algorithm which only outputs hypotheses, $f \in \mathcal{F}$, that are consistent with the training data, (i.e. $\mathbb{E}_{\mathcal{S}}[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})] = 0$)

- Assuming $(0 < \epsilon < 1)$ and $(0 < \delta < 1)$:

$$\mathbb{P}(\textit{a particular } f \in \mathcal{F}, \textit{ with } \mathbb{E}_{\mathcal{D}}[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})] > \epsilon, \textit{ is consistent with } 1 \textit{ training ex.}) < (1 - \epsilon) < e^{-\epsilon}$$

- So:

$$\mathbb{P}(\textit{a particular } f \in \mathcal{F}, \textit{ with } \mathbb{E}_{\mathcal{D}}[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})] > \epsilon, \textit{ is consistent with } n \textit{ training ex.}) < (1 - \epsilon)^n < e^{-\epsilon n}$$

PAC Bound: Finite $\mathcal{F}$, Consistent Learner

- For this to apply to all $f \in \mathcal{F}$ simultaneously, we begin by applying the **Union Bound** to obtain:

$$\mathbb{P}(\mathbb{E}_{\mathcal{D}}[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})] > \epsilon, \text{ for at least one } f \in \mathcal{F}) < |\mathcal{F}| e^{-\epsilon n}$$

- **Theorem:**
  For all $(0 < \epsilon < 1)$, $(0 < \delta < 1)$, all data generating distributions, $\mathcal{D}$, then with probability of at least $(1 - \delta)$:

$$\forall f \in \mathcal{F} \qquad \mathbb{E}_{\mathcal{D}}[\mathcal{E}(f(\mathcal{X}), \mathcal{Y})] \leqslant \frac{1}{n} \ln \left(\frac{|\mathcal{F}|}{\delta} \right)$$

- But what if we wish to access a *more realistic* representation, where our learners can make mistakes?

PAC Bound: Finite $\mathcal{F}$, Agnostic Learner

- An **agnostic learner** is a learning algorithm which is not restricted in the hypotheses it can learn

- Let us recall **Hoeffding's Inequality**:
  Let $\mathcal{Z}_1, \ldots, \mathcal{Z}_n$ be independent random variables, such that each $\mathcal{Z}_i$ is bounded by the interval $[a_i, b_i]$, then for any $\epsilon > 0$:

$$\mathbb{P}\left(\mathbb{E}[\overline{\mathcal{Z}}] - \overline{\mathcal{Z}} \geqslant \epsilon\right) \leqslant \exp\left(-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

PAC Bound: Finite $\mathcal{F}$, Agnostic Learner

- Apply this to i.i.d. $\mathcal{Z}_i$, such that $\mathcal{Z}_i = \mathbb{I}[\mathcal{Y}^{(i)} \neq f(\mathcal{X}^{(i)})]$, means that $(b_i - a_i) = 1$ and:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \geqslant \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[y^{(i)} \neq f(\mathbf{x}^{(i)})] + \epsilon\right) \leqslant e^{-2n\epsilon^2}$$

- And so:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \geqslant \mathbb{E}_{\mathcal{S}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + \epsilon\right) \leqslant e^{-2n\epsilon^2}$$

PAC Bound: Finite $\mathcal{F}$, Agnostic Learner

- For this to hold for all $f \in \mathcal{F}$ simultaneously, we begin by applying the **Union Bound** to obtain:

$$\mathbb{P}\left(\mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \geqslant \mathbb{E}_{\mathcal{S}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + \epsilon, \text{ for at least one } f \in \mathcal{F}\right) \leqslant |\mathcal{F}| e^{-2n\epsilon^2}$$

- **Theorem:**
  For all $(0 < \epsilon < 1)$, $(0 < \delta < 1)$, all data generating distributions, $\mathcal{D}$, then with probability of at least $(1 - \delta)$:

$$\forall f \in \mathcal{F} \qquad \mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \leqslant \mathbb{E}_{\mathcal{S}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + \sqrt{\frac{1}{2n} \ln\left(\frac{|\mathcal{F}|}{\delta}\right)}$$

- But what if we wish to access a *larger* representation?

PAC Bound: Infinite $\mathcal{F}$, Agnostic Learner

- For infinitely large function classes the previous bound becomes trivial because $|\mathcal{F}| \to \infty$
- We need a more subtle way of measuring the **size**, or **functional capacity**, or **complexity** of $\mathcal{F}$
- There are a number of different possible measures, for example:
  - **Rademacher Compexity**
  - **Covering Numbers**
  - **PAC-Bayes**
- Using such a measure we can develop a similar PAC bound to the ones which we've already seen:

PAC Bound: Infinite $\mathcal{F}$, Agnostic Learner

- For example for **misclassification loss**, and the **Rademacher complexity** measure we can derive:

- **Theorem:**
  For all $(0 < \epsilon < 1)$, $(0 < \delta < 1)$, all data generating distributions, $\mathcal{D}$, then with probability of at least $(1 - \delta)$:

$$\forall f \in \mathcal{F} \quad \mathbb{E}_{\mathcal{D}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] \leqslant \mathbb{E}_{\mathcal{S}}[\mathbb{I}[\mathcal{Y} \neq f(\mathcal{X})]] + 2\mathbb{R}(\mathcal{F} \circ \mathcal{S}) + 4\sqrt{\frac{2}{n} \ln\left(\frac{4}{\delta}\right)} \tag{1}$$

- Here $\mathbb{R}(\mathcal{F} \circ \mathcal{S})$ denotes the Rademacher complexity of $\mathcal{F} \circ \mathcal{S} = \{f(\mathbf{x}^{(i)}) | f \in \mathcal{F}\}_{i=1}^{n}$

PAC Bound: Infinite $\mathcal{F}$, Agnostic Learner

- Characterising $\mathbb{R}(\mathcal{F} \circ \mathcal{S})$ is:
  - Beyond the scope of this course!
  - Depends on the training data and on $\mathcal{F}$
  - For some interesting representations a finite quantity...
  - ...even if $\mathcal{F}$ contains an infinite number of functions!

PAC Bound: Model Selection

- So Theorem (1) gives us a worst case upper bound in probability for the generalisation loss for a wide variety of models
- We can calculate this bound for different models ($\mathcal{F}$) characterised by, for example, different hyperparameter settings
- We can perform model selection by picking the model which gives rise to the tightest of these bounds

PAC Bound: Comments

- The PAC approach has advantages:
  - It is not limited to misclassification loss, it can be applied to a variety of other loss functions
  - It allows models to be compared with training data alone, while avoiding overfitting
- However the bounds are often loose:
  - Thus they are often poor estimators of the generalisation loss itself
  - For model selection, PAC bound optimisation tends to underperform cross validation
- But the value of this approach is much broader:
  - It provides a principled basis for understanding why learning algorithms should work
  - And the process of bound formation then optimisation offers a method for principled novel algorithm creation...as we'll see...

Lecture Overview

Model Assessment

- Having chosen a final model, the task of **model assessment** is to estimate the generalisation loss of the model on new data
- As we already saw, the most desirable approach, is to observe model performance on a **validation set**, and then to form a worst case **probabilistic bound** on the generalisation loss
- But in the absence of sufficient data to attempt this, what should we do?

Model Assessment

- The **PAC Bounds** are too loose to be useful for assessment
- The **Cross Validation** loss is a practical and empirically good alternative estimator
- ...While not theoretically well understood, it is widely used in practice

Model Assessment: Metrics

- A related issue is what our metric of efficacy should be
- Learning algorithms are typically designed for canonical loss metrics, e.g.:
  - Mean square error
  - Misclassification loss
- But often we're interested in something different, e.g.:
  - Profitability
  - Quality of Life

Model Assessment: Metrics

- We could design a new algorithm to focus on bespoke metrics
  - But this is hard
- Or we could use our metric of interest in the cross validation phase
  - This can be effective...
  - ...But is a heuristic halfway house which allows us to aim an inadequate algorithm at the right objective

Lecture Overview

Summary

1. Both **Model Selection** and **Model Assessment** involve attempts at characterising the generalisation loss
2. The **Validation Set** approach is a desirable way of tackling these tasks, but requires abundant data
3. The **PAC** approach is a well-motivated alternative but is often practically deficient
4. The **Cross Validation** approach is a widely used (though poorly understood) practical alternative