Academy of PhD Training in Statistics

Statistical Machine Learning

Louis J.M. Aslett (`louis.aslett@durham.ac.uk`)

Supervised Learning

Durham University

This Section

- Gentle introduction to supervised learning
- Formalise the problem setting
- Formalise the approach to learning
- Defining errors and optimal predictors
- Error decompositions
- Consistency
- Model fitting

This Section

- Gentle introduction to supervised learning
- Formalise the problem setting
- Formalise the approach to learning
- Defining errors and optimal predictors
- Error decompositions
- Consistency
- Model fitting

Brace yourself … a ***lot*** *of things to define coming up …*

Notation

- Scalars: $x, x_i, x_{ij}, y_i, \beta_i, \ldots$
- Vectors: $\mathbf{x}, \mathbf{y}, \boldsymbol{\beta}, \ldots$
  - follow the standard convention that all are column vectors
  - transpose $\mathbf{x}^T$ is row vector
  - $\mathbf{x}_i$ indicates a vector, the elements of which are x_{ij}
- Matrices: $\mathbf{X}, \mathbf{Y}, \ldots$
  - matrix transpose: $\mathbf{X}^T$
  - i-th row entries: $\mathbf{x}_i$ (as column vector)
  - j-th column entries: $\mathbf{x}_{\cdot j}$
  - (i, j)-th element: x_{ij}

- Random variables: $X, Y, \varepsilon, \ldots$
  - clear from context whether a random scalar/vector/matrix/…
  - clear from context whether Greek letters are random variables
  - the probability measure associated with a random variable X is π_X.
- Spaces: $\mathcal{X}, \mathcal{Y}, \mathbb{R}, \mathbb{Z}, \mathbb{R}^d = \underbrace{\mathbb{R} \times \cdots \times \mathbb{R}}_{d \text{ times}}, \ldots$
- Estimator: denoted by a hat, $\hat{f}(\cdot), \hat{\boldsymbol{\beta}}, \ldots$
- Functions:
  - $\mathbb{1}\{A\}$ is the indicator function for A being true.
  - if $f(x)$ has vector valued output, then $f_j(x)$ denotes the j-th component of the output.
  - where necessary an arbitrary function, $f(x)$, will be distinguished from a probability density function (pdf), $f_X(x)$, by the presence or absence of a random variable subscript.
  - the cumulative distribution function (cdf) is denoted $F_X(x)$.

- Other:
  - $A := B$ reads "A is defined to be B".
  - For a finite set $\mathcal{I}$, $|\mathcal{I}|$ denotes the cardinality of the set.

Note: we avoid excessive formality and write $\min \ldots$, implicitly assuming minimum exists; likewise, $\mathbb{E}[\;]$ makes an implicit assumption that it exists.

Problem setting (I)

The most common supervised machine learning problems fall broadly under three types (Vapnik, 1998):

- **Regression** models a *quantitative* outcome.
  - What value is a house based on geographic/house information;
  - How long until a patient is discharged from hospital?
- **Classification** models a *qualitative* outcome.
  - Medics predicting a disease from test results;
  - Is the email just sent to my address spam?
  - Bank predicting if borrower will default;
  - Identifying a number from image of handwritten value.
- **Density estimation** models a full *probability distribution*.

Problem setting (II)

Premise: we have access to a set of n observations of

- *features* / *predictors* from some space $\mathcal{X}$, eg:
  - $\mathcal{X} \subset \mathbb{R}^d$
  - or, $\mathcal{X}$ can be a tensor in deep learning
- and corresponding *outcomes* / *responses* / *targets* from some space $\mathcal{Y}$, eg:
  - $\mathcal{Y} \subset \mathbb{R} \implies$ regression;
  - or, $\mathcal{Y} = \{1, \dots, g\}$ where $g \geq 2 \implies$ classification;
  - or, for $g = 2$ often take $\mathcal{Y} = \{0, 1\}$
  - or, $\mathcal{Y}$ can be a tensor in deep learning

Problem setting (II)

Premise: we have access to a set of n observations of

- *features* / *predictors* from some space $\mathcal{X}$, eg:
  - $\mathcal{X} \subset \mathbb{R}^d$
  - or, $\mathcal{X}$ can be a tensor in deep learning
- and corresponding *outcomes* / *responses* / *targets* from some space $\mathcal{Y}$, eg:
  - $\mathcal{Y} \subset \mathbb{R} \implies$ regression;
  - or, $\mathcal{Y} = \{1, \dots, g\}$ where $g \geq 2 \implies$ classification;
  - or, for $g = 2$ often take $\mathcal{Y} = \{0, 1\}$
  - or, $\mathcal{Y}$ can be a tensor in deep learning

Dataset is $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\} \subset (\mathcal{X} \times \mathcal{Y})^n$, where $\mathbf{x}_i$ is a vector of length d,

$$\mathbf{x}_i = (x_{i1}, \dots, x_{id})^T \in \mathcal{X}$$

All observations of single feature $\mathbf{x}_{\cdot j}$, $\mathbf{x}_{\cdot j} = (x_{1j}, \dots, x_{nj})^T$

Problem setting: objective

Objective is to learn relationship between features and response:

Regression: $Y = f(X) + \varepsilon$

ε is random error term, assumed mean zero.

Problem setting: objective

Objective is to learn relationship between features and response:

Regression: $Y = f(X) + \varepsilon$

ε is random error term, assumed mean zero.

Classification: $(Y \mid X = \mathbf{x}) \sim \text{Categorical}\big((p_1, \dots, p_g) = f(\mathbf{x})\big)$

where $p_i = \mathbb{P}(Y = i \mid X = \mathbf{x})$

Problem setting: objective

Objective is to learn relationship between features and response:

Regression: $Y = f(X) + \varepsilon$

ε is random error term, assumed mean zero.

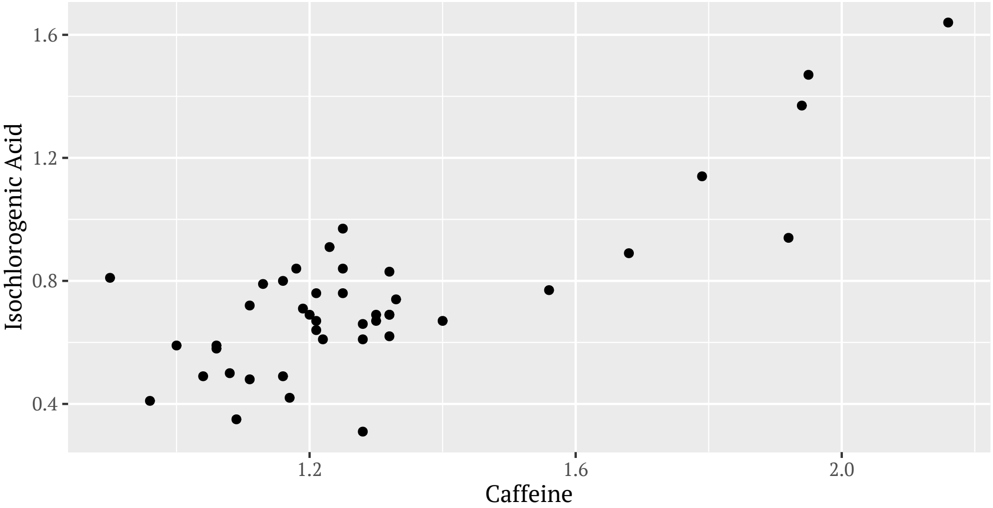
Classification: $(Y \mid X = \mathbf{x}) \sim \text{Categorical}\big((p_1, \dots, p_g) = f(\mathbf{x})\big)$

where $p_i = \mathbb{P}(Y = i \mid X = \mathbf{x})$

Density Estimation: $(Y \mid X = x) \sim \mathbb{P}(Y \mid X = x)$

possibly by jointly modelling $\mathbb{P}(X, Y)$

“A mathematician is a machine for turning coffee into theorems”



Example 2D regression

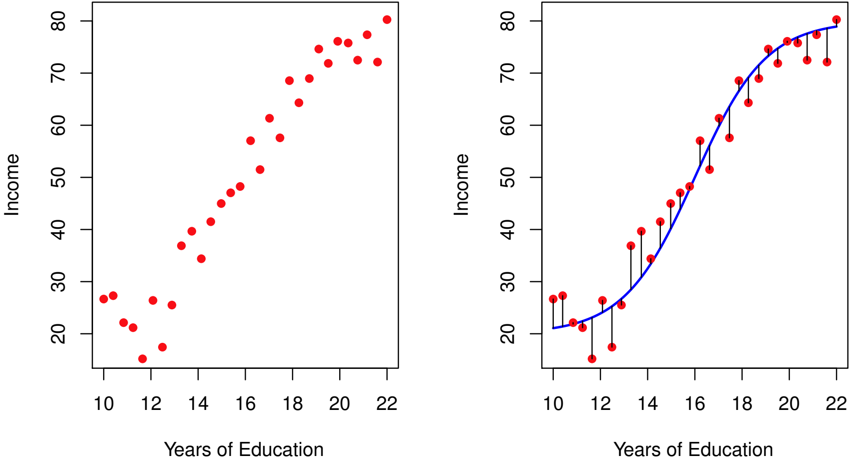


Figure 1: From “An Introduction to Statistical Learning”.

Example 3D regression

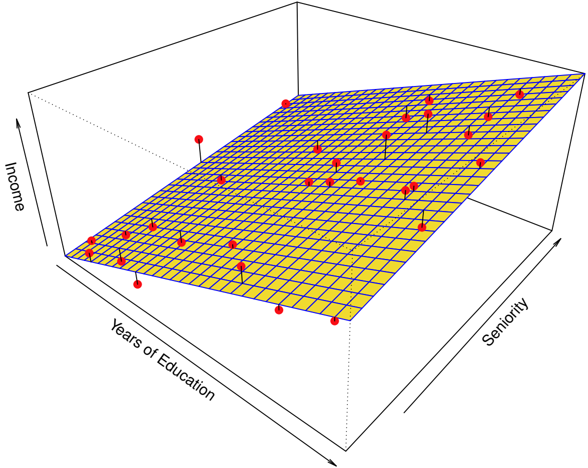


Figure 2: From “An Introduction to Statistical Learning”.

Example 3D regression

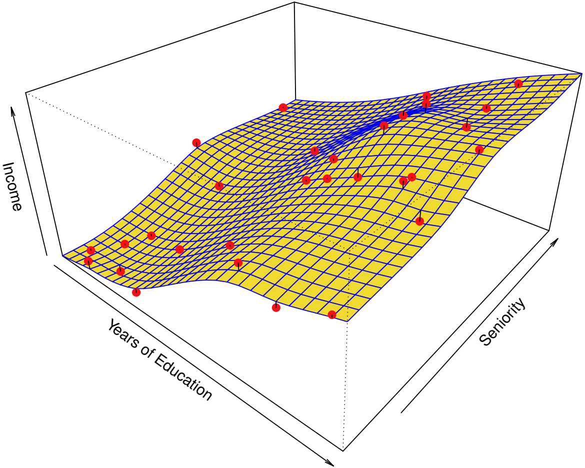


Figure 2: From “An Introduction to Statistical Learning”.

Example 3D regression

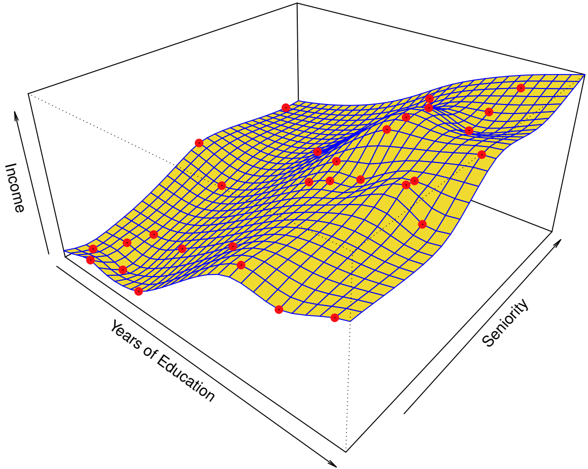
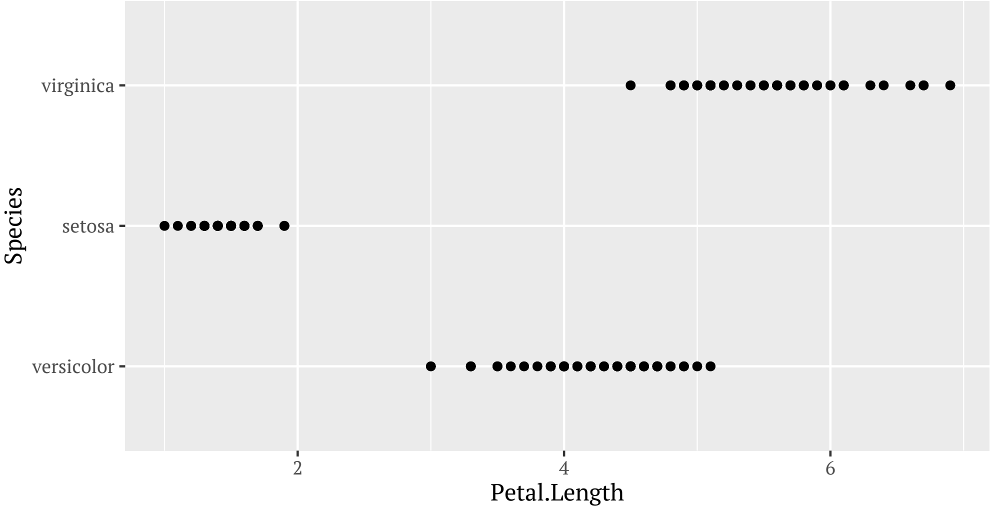
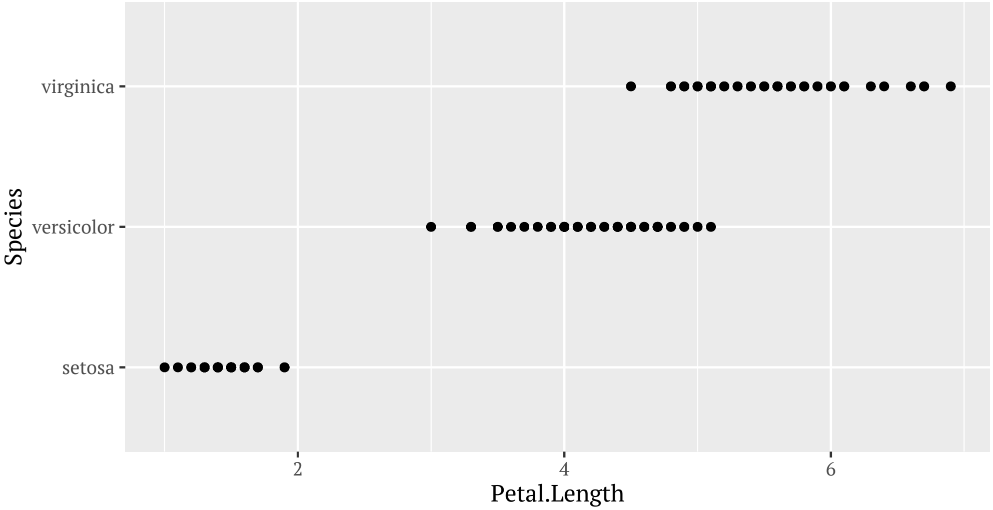


Figure 2: From “An Introduction to Statistical Learning”.

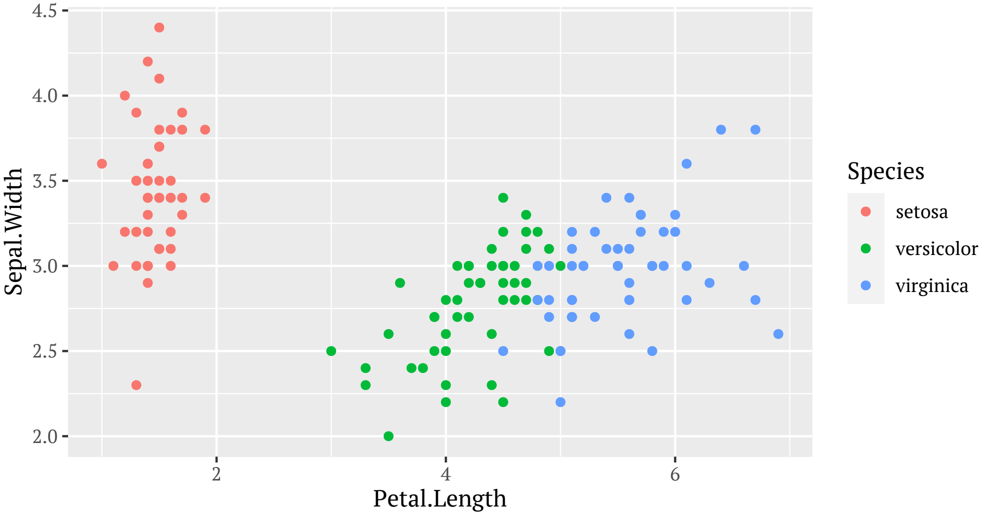
Example classification (I)



Example classification (II)



Example classification (III)



Problem setting: summary

- Focus is heavily on prediction and predictive accuracy for future observations
- Assume in future we have access to $\mathbf{x}$, but not y
- The “model” setting is very general, we usually directly tackle estimation of $\hat{f}$

Problem setting: summary

- Focus is heavily on prediction and predictive accuracy for future observations
- Assume in future we have access to $\mathbf{x}$, but not y
- The "model" setting is very general, we usually directly tackle estimation of $\hat{f}$

Important aside: we are going to be interested in the *random inputs* scenario, not the *fixed inputs* one that is often studied in classical statistics setting.

Loss functions

Given fitted model $\hat{f}(\cdot)$ and newly observed feature vector $\mathbf{x}$, denote the prediction $\hat{y} := g_{\hat{f}}(\mathbf{x})$.

We want to minimise the *loss* we suffer when predicting $\hat{y}$ and actually observing y. To do so, define a loss function,

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$$

which measures the discrepancy between prediction and reality.

Loss functions

Given fitted model $\hat{f}(\cdot)$ and newly observed feature vector $\mathbf{x}$, denote the prediction $\hat{y} := g_{\hat{f}}(\mathbf{x})$.

We want to minimise the *loss* we suffer when predicting $\hat{y}$ and actually observing y. To do so, define a loss function,

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$$

which measures the discrepancy between prediction and reality.

For classification, often

$$\mathcal{L} : \mathcal{Y} \times [0, 1]^g \to [0, \infty)$$

Loss functions: regression

- Square loss, $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$
- Absolute loss, $\mathcal{L}(y, \hat{y}) = |y - \hat{y}|$
- Quantile loss, (sometimes called pinball loss),

$$\mathcal{L}(y, \hat{y}) = \begin{cases} (1 - \alpha)(\hat{y} - y) & \text{if } y \leq \hat{y} \\ \alpha(y - \hat{y}) & \text{if } y > \hat{y} \end{cases}$$

where $\alpha \in (0, 1)$ is the target quantile.

Loss functions: classification

- 0-1 loss, $\mathcal{L}(y, \hat{y}) = \mathbb{1}\{y \neq \hat{y}\}$
- Cross entropy loss, $\mathcal{L}(y, \hat{\mathbf{p}}) = -\sum_{j=1}^{g} \mathbb{1}\{y = j\} \log \hat{p}_j = -\log \hat{p}_y$
- Brier score loss, $\mathcal{L}(y, \hat{\mathbf{p}}) = \sum_{j=1}^{g} (\mathbb{1}\{y = j\} - \hat{p}_j)^2$
- Exponential loss (binary+scoring setting, $y \in \{-1, 1\}$), $\mathcal{L}(y, \hat{f}(\mathbf{x})) = \exp(-y\hat{f}(\mathbf{x}))$

Generalisation error

The *generalisation error* (sometimes called *generalisation risk*) of a model $\hat{f}(\cdot)$, with respect to a loss $\mathcal{L}$, is the expected loss of a future prediction $g_{\hat{f}}(\cdot)$ with respect to the true data generating measure π_{XY},

$$\mathcal{E}(\hat{f}) := \mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X))\right]$$

Generalisation error

The *generalisation error* (sometimes called *generalisation risk*) of a model $\hat{f}(\cdot)$, with respect to a loss $\mathcal{L}$, is the expected loss of a future prediction $g_{\hat{f}}(\cdot)$ with respect to the true data generating measure π_{XY},

$$\mathcal{E}(\hat{f}) := \mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X))\right]$$

Note: $\hat{f}$ is *not* a random variable … assumes a fixed fitted model already.

Estimated generalisation error

The *estimated generalisation error* based on dataset $\mathcal{D} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$, where $(\mathbf{x}_i, y_i) \overset{iid}{\sim} \pi_{XY}$ is,

$$\hat{\mathcal{E}}_{\mathcal{D}}(\hat{f}) := \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i)) \approx \mathcal{E}(\hat{f})$$

Estimated generalisation error

The *estimated generalisation error* based on dataset $\mathcal{D} = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m))$, where $(\mathbf{x}_i, y_i) \overset{iid}{\sim} \pi_{XY}$ is,

$$\hat{\mathcal{E}}_{\mathcal{D}}(\hat{f}) := \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i)) \approx \mathcal{E}(\hat{f})$$

$$\text{Var}\left(\hat{\mathcal{E}}_{\mathcal{D}}(\hat{f})\right) \approx \frac{1}{m(m-1)} \sum_{i=1}^{m} \left(\mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i)) - \hat{\mathcal{E}}_{\mathcal{D}}(\hat{f})\right)^2$$

Estimated generalisation error

The *estimated generalisation error* based on dataset $\mathcal{D} = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m))$, where $(\mathbf{x}_i, y_i) \overset{iid}{\sim} \pi_{XY}$ is,

$$\hat{\mathcal{E}}_{\mathcal{D}}(\hat{f}) := \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i)) \approx \mathcal{E}(\hat{f})$$

$$\text{Var}\left(\hat{\mathcal{E}}_{\mathcal{D}}(\hat{f})\right) \approx \frac{1}{m(m-1)} \sum_{i=1}^{m} \left(\mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i)) - \hat{\mathcal{E}}_{\mathcal{D}}(\hat{f})\right)^2$$

- $\mathcal{D}$ = data used to fit $\hat{f} \implies$ *training/apparent error*
- $\mathcal{D}$ = different iid data from $\pi_{XY} \implies$ *test error*

Example

Example in notes

Training/apparent error

Problems with training/apparent error

1. estimate is constrained to the same predictor/feature values as in the data that was used to fit the model;
2. model fitting specifically adapted to the particular responses in the training data, so the error not representative of future responses, even when made at same predictor values.

So …

- training/apparent error is biased (point 2); and
- estimating subtly different quantity (point 1)! In-sample fixed inputs error:

$$\text{Err} := \mathbb{E}_{Y \mid X = \mathbf{x}_i} \left[\mathcal{L}(Y, g_{\hat{f}}(\mathbf{x}_i)) \right] \qquad \widehat{\text{Err}} := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, g_{\hat{f}}(\mathbf{x}_i))$$

What to predict?

Note, by total law of expectation:

$$\begin{aligned}\mathcal{E}(f) &= \mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X))\right] \\ &= \mathbb{E}_{X}\left[\mathbb{E}_{Y \mid X}\left[\mathcal{L}(Y, g_f(X)) \mid X = \mathbf{x}\right]\right]\end{aligned}$$

$\therefore$ pointwise solution to minimise generalisation error means we should ideally choose for our prediction function $g_f(X)$ to predict:

$$g^\star(\mathbf{x}) := \arg\min_{z \in \mathcal{Y}} \mathbb{E}_{Y \mid X}\left[\mathcal{L}(Y, z) \mid X = \mathbf{x}\right]$$

This is the so-called *Bayes predictor*.

Bayes predictor: 0-1 loss

$$\begin{aligned}
g^\star(\mathbf{x}) &= \arg\min_{z \in \mathcal{Y}} \mathbb{E}\left[\mathbb{1}\{Y \neq z\} \,|\, X = \mathbf{x}\right] \\
&= \arg\min_{z \in \mathcal{Y}} \mathbb{P}\left(Y \neq z \,|\, X = \mathbf{x}\right) \\
&= \arg\min_{z \in \mathcal{Y}} 1 - \mathbb{P}\left(Y = z \,|\, X = \mathbf{x}\right) \\
&= \arg\max_{z \in \mathcal{Y}} \mathbb{P}\left(Y = z \,|\, X = \mathbf{x}\right)
\end{aligned}$$

Bayes predictor: 0-1 loss

$$\begin{aligned} g^\star(\mathbf{x}) &= \arg\min_{z\in\mathcal{Y}} \mathbb{E}\left[\mathbb{1}\{Y \neq z\} \,|\, X = \mathbf{x}\right] \\ &= \arg\min_{z\in\mathcal{Y}} \mathbb{P}\left(Y \neq z \,|\, X = \mathbf{x}\right) \\ &= \arg\min_{z\in\mathcal{Y}} 1 - \mathbb{P}\left(Y = z \,|\, X = \mathbf{x}\right) \\ &= \arg\max_{z\in\mathcal{Y}} \mathbb{P}\left(Y = z \,|\, X = \mathbf{x}\right) \end{aligned}$$

$$\therefore \quad (Y \,|\, X = \mathbf{x}) \sim \text{Categorical}\big((p_1, \dots, p_g) = f(\mathbf{x})\big)$$

$$\implies g^\star(\mathbf{x}) = \arg \max_{j\in\{1,\dots,g\}} f_j(\mathbf{x})$$

Bayes predictor: 0-1 loss

$$\begin{aligned}
g^\star(\mathbf{x}) &= \arg\min_{z \in \mathcal{Y}} \mathbb{E}\left[\mathbb{1}\{Y \neq z\} \mid X = \mathbf{x}\right] \\
&= \arg\min_{z \in \mathcal{Y}} \mathbb{P}\left(Y \neq z \mid X = \mathbf{x}\right) \\
&= \arg\min_{z \in \mathcal{Y}} 1 - \mathbb{P}\left(Y = z \mid X = \mathbf{x}\right) \\
&= \arg\max_{z \in \mathcal{Y}} \mathbb{P}\left(Y = z \mid X = \mathbf{x}\right)
\end{aligned}$$

$$\therefore \quad (Y \mid X = \mathbf{x}) \sim \text{Categorical}\big((p_1, \dots, p_g) = f(\mathbf{x})\big)$$

$$\implies \quad g^\star(\mathbf{x}) = \arg\max_{j \in \{1,\dots,g\}} f_j(\mathbf{x})$$

$$\implies \quad g_{\hat{f}}(\mathbf{x}) = \arg\max_{j \in \{1,\dots,g\}} \hat{f}_j(\mathbf{x})$$

Bayes predictor: square loss

Similarly,

$$\begin{aligned} g^\star(\mathbf{x}) &= \arg\min_{z\in\mathcal{Y}} \mathbb{E}\left[(Y-z)^2 \mid X=\mathbf{x}\right] \\ &= \arg\min_{z\in\mathcal{Y}} \mathbb{E}\left[((Y-\mathbb{E}\left[Y \mid X=\mathbf{x}\right]) + (\mathbb{E}\left[Y \mid X=\mathbf{x}\right]-z))^2 \mid X=\mathbf{x}\right] \\ &= \mathbb{E}\left[Y \mid X=\mathbf{x}\right] \end{aligned}$$

$$\begin{aligned} \therefore \quad & Y = f(X) + \varepsilon,\ \varepsilon \text{ zero mean} \\ \implies & g^\star(\mathbf{x}) = f(\mathbf{x}) \\ \implies & g_{\hat{f}}(\mathbf{x}) = \hat{f}(\mathbf{x}) \end{aligned}$$

Bayes error & excess risk

The *Bayes error* is the generalisation error which arises when using the Bayes predictor,

$$\mathcal{E}^\star = \mathbb{E}_X \left[\inf_{z \in \mathcal{Y}} \mathbb{E}_{Y \,|\, X} \left[\mathcal{L}(Y, z) \,|\, X = \mathbf{x} \right] \right]$$

Best performance one could hope to achieve!

Bayes error & excess risk

The *Bayes error* is the generalisation error which arises when using the Bayes predictor,

$$\mathcal{E}^\star = \mathbb{E}_X \left[\inf_{z \in \mathcal{Y}} \mathbb{E}_{Y \mid X} \left[\mathcal{L}(Y, z) \mid X = \mathbf{x} \right] \right]$$

Best performance one could hope to achieve! *Never achievable!*

Bayes error & excess risk

The *Bayes error* is the generalisation error which arises when using the Bayes predictor,

$$\mathcal{E}^\star = \mathbb{E}_X \left[\inf_{z \in \mathcal{Y}} \mathbb{E}_{Y \mid X} \left[\mathcal{L}(Y, z) \mid X = \mathbf{x} \right] \right]$$

Best performance one could hope to achieve! *Never achievable!*

The *excess risk* is the increase in generalisation error above the Bayes error suffered by a given fitted model $\hat{f}$, that is, $\mathcal{E}(\hat{f}) - \mathcal{E}^\star$.

Error decompositions

$$\mathcal{E}(\hat{f})$$

Error decompositions

$$\mathcal{E}(\hat{f}) = \mathcal{E}(\hat{f})$$

Error decompositions

$$\mathcal{E}(\hat{f}) = \mathcal{E}(\hat{f}) - \mathcal{E}^{\star} + \mathcal{E}^{\star}$$

Error decompositions

$$
\begin{aligned}
\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\
&= \mathcal{E}(\hat{f}) \qquad\qquad\qquad\qquad - \mathcal{E}^\star \;+ \qquad \mathcal{E}^\star
\end{aligned}
$$

Error decompositions

$$
\begin{aligned}
\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\
&= \mathcal{E}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{E}(f) + \inf_{f \in \mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star \quad + \qquad \mathcal{E}^\star
\end{aligned}
$$

Error decompositions

$$\begin{aligned}\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\ &= \underbrace{\mathcal{E}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{E}(f)}_{\text{estimation error}} + \inf_{f\in\mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star \quad + \qquad \mathcal{E}^\star\end{aligned}$$

Error decompositions

$$\begin{aligned}\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\ &= \underbrace{\mathcal{E}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{E}(f)}_{\text{estimation error}} + \underbrace{\inf_{f\in\mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star}_{\text{approximation error}} + \qquad \mathcal{E}^\star\end{aligned}$$

Error decompositions

$$
\begin{aligned}
\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\
&= \underbrace{\underbrace{\mathcal{E}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{E}(f)}_{\text{estimation error}} + \underbrace{\inf_{f\in\mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star}_{\text{approximation error}}}_{\text{reducible error } (\equiv\text{excess risk})} + \qquad \mathcal{E}^\star
\end{aligned}
$$

Error decompositions

$$
\begin{aligned}
\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\
&= \underbrace{\underbrace{\mathcal{E}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{E}(f)}_{\text{estimation error}} + \underbrace{\inf_{f \in \mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star}_{\text{approximation error}}}_{\text{reducible error } (\equiv \text{excess risk})} + \underbrace{\mathcal{E}^\star}_{\text{irreducible error}}
\end{aligned}
$$

Error decompositions

$$
\begin{aligned}
\mathcal{E}(\hat{f}) &= \mathcal{E}(\hat{f}) - \mathcal{E}^\star + \mathcal{E}^\star \\
&= \underbrace{\underbrace{\mathcal{E}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{E}(f)}_{\text{estimation error}} + \underbrace{\inf_{f\in\mathcal{F}} \mathcal{E}(f) - \mathcal{E}^\star}_{\text{approximation error}}}_{\text{reducible error } (\equiv\text{excess risk})} + \underbrace{\mathcal{E}^\star}_{\text{irreducible error}}
\end{aligned}
$$

$$
\mathbb{E}[(Y - \hat{f}(X))^2] = \underbrace{\mathbb{E}\left[\left(f(X) - \hat{f}(X)\right)^2\right]}_{\text{reducible error}} + \underbrace{\mathrm{Var}(\varepsilon)}_{\text{irreducible error}}
$$

Recap

Take a breath:

- Loss functions to assess quality of prediction

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
    - estimated via test error
    - training/apparent error leads to in-sample fixed inputs error

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
    - estimated via test error
    - training/apparent error leads to in-sample fixed inputs error
- Bayes predictor shows what to predict given our model to minimise error

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
    - estimated via test error
    - training/apparent error leads to in-sample fixed inputs error
- Bayes predictor shows what to predict given our model to minimise error
- Bayes error is best we could do

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
    - estimated via test error
    - training/apparent error leads to in-sample fixed inputs error
- Bayes predictor shows what to predict given our model to minimise error
- Bayes error is best we could do
- Excess risk is how much worse than the best we actually do!

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
  - estimated via test error
  - training/apparent error leads to in-sample fixed inputs error
- Bayes predictor shows what to predict given our model to minimise error
- Bayes error is best we could do
- Excess risk is how much worse than the best we actually do!
- Generalisation error can be decomposed into reducible (estimation+approximation) error and irreducible error

Recap

Take a breath:

- Loss functions to assess quality of prediction
- Generalisation error to assess a fixed fitted model
    - estimated via test error
    - training/apparent error leads to in-sample fixed inputs error
- Bayes predictor shows what to predict given our model to minimise error
- Bayes error is best we could do
- Excess risk is how much worse than the best we actually do!
- Generalisation error can be decomposed into reducible (estimation+approximation) error and irreducible error

But … so far, everything predicated on fixed, already fitted $\hat{f}$!

Training data

So far, model dependency on data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ "hidden" to simplify notation and data fixed realisation.

$\hat{f}(\mathbf{x})$ could be written $\hat{f}(\mathbf{x} \,|\, \mathcal{D})$ to stress it is a model fitted to that data … change the data, model changes (obviously!)

Training data

So far, model dependency on data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ “hidden” to simplify notation and data fixed realisation.

$\hat{f}(\mathbf{x})$ could be written $\hat{f}(\mathbf{x} \,|\, \mathcal{D})$ to stress it is a model fitted to that data … change the data, model changes (obviously!)

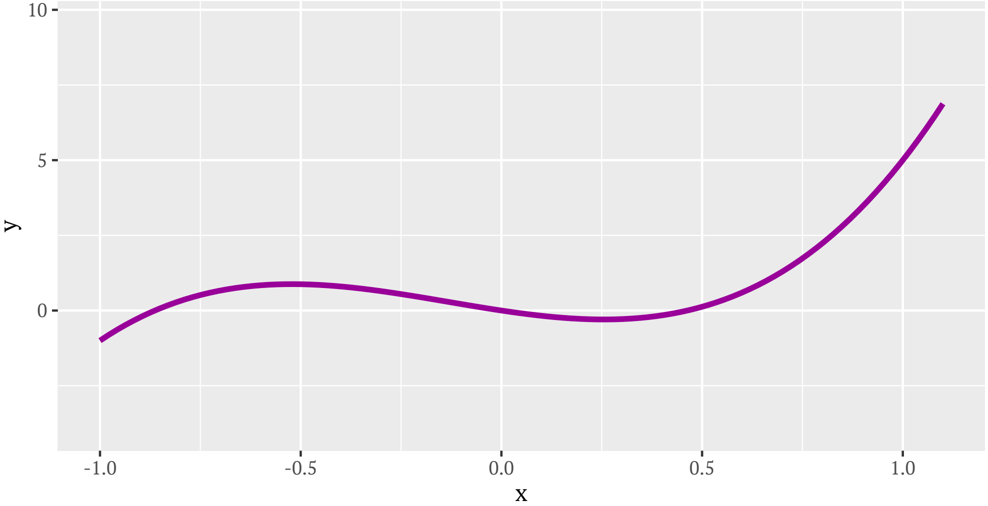
$$D_n := ((X_1, Y_1), \dots, (X_n, Y_n))$$

defined to be the random variable for n observations from joint distribution

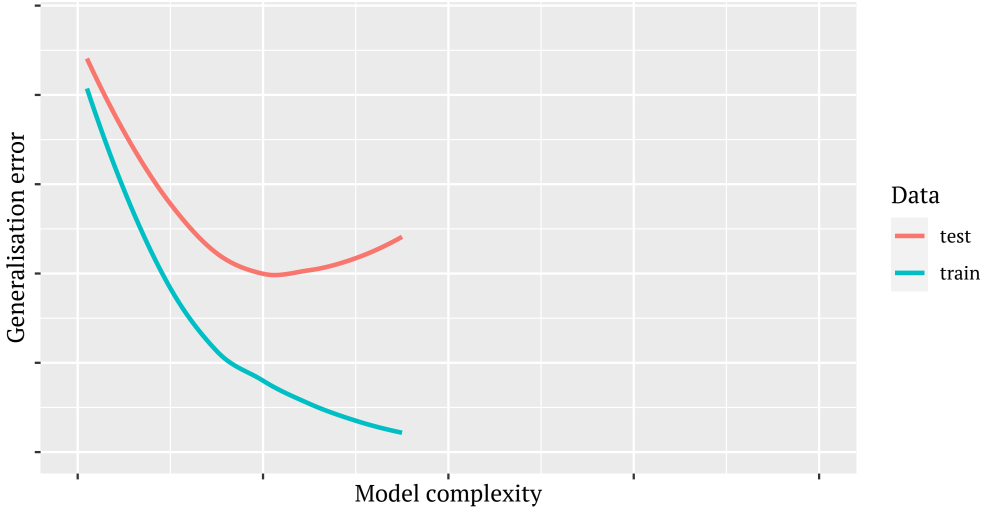
$$\pi_{XY}^n := \underbrace{\pi_{XY} \times \cdots \times \pi_{XY}}_{n \text{ times}}$$

So, $\hat{f}(\mathbf{x} \,|\, \mathcal{D}_n)$ is realisation of $\hat{f}(\mathbf{x} \,|\, D_n)$

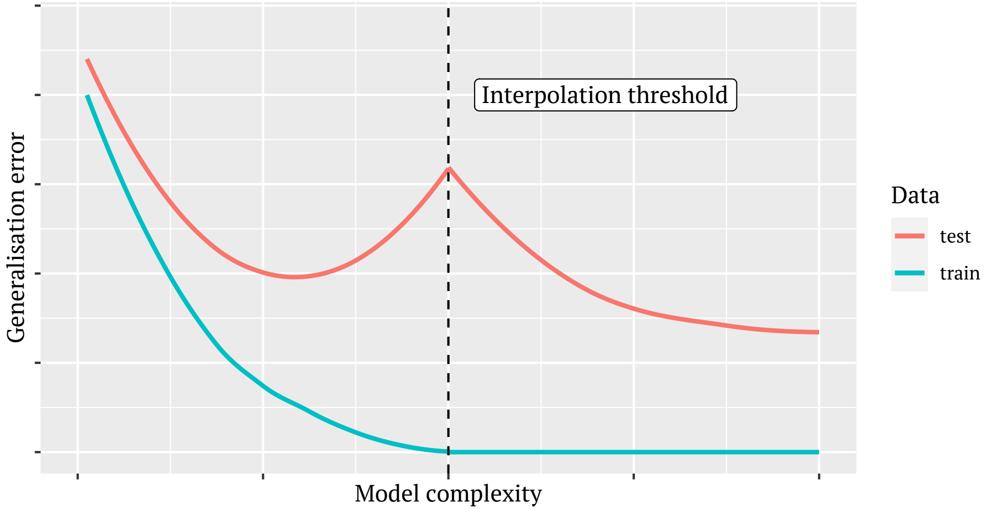
$X \sim \text{Unif}(-1, 1.1), (Y \mid X = x) \sim \text{N}(\mu = 5x^3 + 2x^2 - 2x, \sigma = 1)$



Generalisation error and model complexity



The *double descent* phenomenon



Expected (prediction) error

The *expected error* of a learning algorithm which learns $\hat{f} \in \mathcal{F}$ given data sample $D_n \sim \pi_{XY}^n$ is,

$$\bar{\mathcal{E}}_n := \mathbb{E}_{D_n}\left[\mathcal{E}(\hat{f})\right] = \mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right]$$

Expected (prediction) error

The *expected error* of a learning algorithm which learns $\hat{f} \in \mathcal{F}$ given data sample $D_n \sim \pi_{XY}^n$ is,

$$\bar{\mathcal{E}}_n := \mathbb{E}_{D_n}\left[\mathcal{E}(\hat{f})\right] = \mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right]$$

Sometimes more interested in the *expected* *<u>prediction</u>* *error* of a learning algorithm at a particular predictor value $X = \mathbf{x}$,

$$\bar{\mathcal{E}}_n(\mathbf{x}) := \mathbb{E}_{D_n}\left[\mathbb{E}_{Y \mid X=\mathbf{x}}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right]$$

where now the inner expectation is conditioned on the predictor.

Consistency

A learning algorithm is *consistent* for π_{XY} if it is asymptotically Bayes error efficient.

i.e. if the expected error converges to the Bayes error in the limit as the sample size grows,

$$\mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right] \to \mathcal{E}^\star \text{ as } n \to \infty$$

Consistency

A learning algorithm is *consistent* for π_{XY} if it is asymptotically Bayes error efficient.

i.e. if the expected error converges to the Bayes error in the limit as the sample size grows,

$$\mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right] \to \mathcal{E}^\star \text{ as } n \to \infty$$

A learning algorithm is *universally consistent* if it is consistent for *all* data generating measures π_{XY}.

Consistency

A learning algorithm is *consistent* for π_{XY} if it is asymptotically Bayes error efficient.

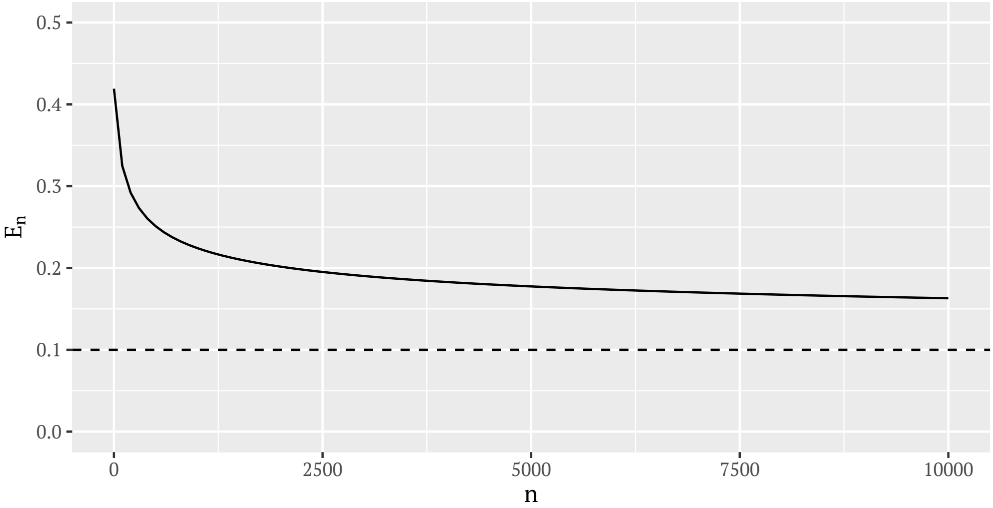
i.e. if the expected error converges to the Bayes error in the limit as the sample size grows,

$$\mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\mathcal{L}(Y, g_{\hat{f}}(X \mid D_n))\right]\right] \to \mathcal{E}^\star \text{ as } n \to \infty$$

A learning algorithm is *universally consistent* if it is consistent for *all* data generating measures π_{XY}.

The catch: universally consistent methods can be very data hungry, so often *under*perform non-universally consistent methods in finite data regime!

Learning curves



To err is human …

Table 1: *Relationship between types of error*

	Fixed inputs	**Random inputs**
Fixed training set	$\mathrm{Err}(\cdot)$	$\mathcal{E}(\cdot)$
Random training set	$\star$	$\bar{\mathcal{E}}_n(\cdot)$

Full error decomposition (I)

For square loss, taking expectations wrt D_n of the earlier decomposition:

$$\bar{\mathcal{E}}_n = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(Y - \hat{f}(X)\right)^2\right] = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] + \mathbb{E}_{D_n}\mathrm{Var}_{XY}(\varepsilon)$$

Full error decomposition (I)

For square loss, taking expectations wrt D_n of the earlier decomposition:

$$\bar{\mathcal{E}}_n = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(Y - \hat{f}(X)\right)^2\right] = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] + \mathbb{E}_{D_n}\mathrm{Var}_{XY}(\varepsilon)$$

$$\mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] = \underbrace{\mathbb{E}_{XY}\mathbb{E}_{D_n}}_{\text{Fubini-Tonelli Theorem}}\left[\left(f(X) - \hat{f}(X)\right)^2\right]$$

Full error decomposition (I)

For square loss, taking expectations wrt D_n of the earlier decomposition:

$$\bar{\mathcal{E}}_n = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(Y - \hat{f}(X)\right)^2\right] = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] + \mathbb{E}_{D_n}\mathrm{Var}_{XY}(\varepsilon)$$

$$\mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] = \underbrace{\mathbb{E}_{XY}\mathbb{E}_{D_n}}_{\text{Fubini-Tonelli Theorem}}\left[\left(f(X) - \hat{f}(X)\right)^2\right]$$

$$= \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\left(\left(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\right) + \left(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\right)\right)^2\right] \quad \text{just } \pm \text{ same term}$$

Full error decomposition (I)

For square loss, taking expectations wrt D_n of the earlier decomposition:

$$\bar{\mathcal{E}}_n = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(Y - \hat{f}(X)\right)^2\right] = \mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] + \mathbb{E}_{D_n}\mathrm{Var}_{XY}(\varepsilon)$$

$$\begin{aligned}
\mathbb{E}_{D_n}\mathbb{E}_{XY}\left[\left(f(X) - \hat{f}(X)\right)^2\right] &= \underbrace{\mathbb{E}_{XY}\mathbb{E}_{D_n}}_{\text{Fubini-Tonelli Theorem}}\left[\left(f(X) - \hat{f}(X)\right)^2\right] \\
&= \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\left(\left(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\right) + \left(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\right)\right)^2\right] \quad \text{just } \pm \text{ same term} \\
&= \mathbb{E}_{XY}\mathbb{E}_{D_n}\Bigg[\left(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\right)^2 + \left(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\right)^2 \\
&\qquad + 2\left(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\right)\left(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\right)\Bigg]
\end{aligned}$$

Full error decomposition (II)

$$\mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\underbrace{2\left(f(X)-\mathbb{E}_{D_n}\hat{f}(X)\right)}_{\text{constant wrt training data}}\left(\mathbb{E}_{D_n}\hat{f}(X)-\hat{f}(X)\right)\right]$$

Full error decomposition (II)

$$\begin{aligned}
&\mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\underbrace{2\left(f(X)-\mathbb{E}_{D_n}\hat{f}(X)\right)}_{\text{constant wrt training data}}\left(\mathbb{E}_{D_n}\hat{f}(X)-\hat{f}(X)\right)\right]\\
&=2\mathbb{E}_{XY}\left[\left(f(X)-\mathbb{E}_{D_n}\hat{f}(X)\right)\underbrace{\mathbb{E}_{D_n}\left[\mathbb{E}_{D_n}\hat{f}(X)-\hat{f}(X)\right]}_{=\mathbb{E}_{D_n}\hat{f}(X)-\mathbb{E}_{D_n}\hat{f}(X)=0}\right]
\end{aligned}$$

Full error decomposition (II)

$$
\begin{aligned}
&\mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\underbrace{2\left(f(X)-\mathbb{E}_{D_n}\hat{f}(X)\right)}_{\text{constant wrt training data}}\left(\mathbb{E}_{D_n}\hat{f}(X)-\hat{f}(X)\right)\right]\\
&=2\mathbb{E}_{XY}\left[\left(f(X)-\mathbb{E}_{D_n}\hat{f}(X)\right)\underbrace{\mathbb{E}_{D_n}\left[\mathbb{E}_{D_n}\hat{f}(X)-\hat{f}(X)\right]}_{=\mathbb{E}_{D_n}\hat{f}(X)-\mathbb{E}_{D_n}\hat{f}(X)=0}\right]\\
&=0
\end{aligned}
$$

Full error decomposition (III)

$$
\begin{aligned}
\bar{\mathcal{E}}_n = \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\Big(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\Big)^2\right] & & \text{squared bias of model} \\
+ \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\Big(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\Big)^2\right] & = \mathbb{E}_{XY}\mathrm{Var}_{D_n}\hat{f}(X) & \text{variance of model fit} \\
+ \mathrm{Var}_{XY}(\varepsilon) & & \text{irreducible error}
\end{aligned}
$$

Full error decomposition (III)

$$\begin{aligned}
\bar{\mathcal{E}}_n &= \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\left(f(X) - \mathbb{E}_{D_n}\hat{f}(X)\right)^2\right] && \text{squared bias of model} \\
&+ \mathbb{E}_{XY}\mathbb{E}_{D_n}\left[\left(\mathbb{E}_{D_n}\hat{f}(X) - \hat{f}(X)\right)^2\right] \quad = \mathbb{E}_{XY}\mathrm{Var}_{D_n}\hat{f}(X) && \text{variance of model fit} \\
&+ \mathrm{Var}_{XY}(\varepsilon) && \text{irreducible error}
\end{aligned}$$

This is a lot more complicated in the non-square loss case, but similar decompositions can be derived.

Bias, variance and irreducible error

Bias

How well could my model, $\hat{f}(\cdot)$, possibly approximate the true deterministic part of the relationship, $f(\cdot)$, assuming I could see as much data as I like?

Variance

How sensitive is the fitting of my model, $\hat{f}(\cdot)$, to the actual finite amount of data I have to learn from?

Irreducible error

How much “true” randomness is there inherent to the problem which we could never hope to deterministically model?

Bias, variance and irreducible error

Bias

How well could my model, $\hat{f}(\cdot)$, possibly approximate the true deterministic part of the relationship, $f(\cdot)$, assuming I could see as much data as I like?

Variance

How sensitive is the fitting of my model, $\hat{f}(\cdot)$, to the actual finite amount of data I have to learn from?

Irreducible error

How much "true" randomness is there inherent to the problem which we could never hope to deterministically model?

Maybe just need very *flexible* models than can be *accurately* fitted to problems with little *inherent randomness*? As we saw, these things all interact in a difficult way.

Model fitting

Broadly three categories of ML model:

1. Full probabilistic model;
2. Parametric family without explicit probabilistic structure;
3. Local method constructing non-parametric empirical estimator;

Model fitting

Broadly three categories of ML model:

1. Full probabilistic model;

   → use your statistical prowess!

2. Parametric family without explicit probabilistic structure;

3. Local method constructing non-parametric empirical estimator;

Model fitting

Broadly three categories of ML model:

1. Full probabilistic model;

   → use your statistical prowess!

2. Parametric family without explicit probabilistic structure;

   → Empirical Risk Minimisation (ERM)

3. Local method constructing non-parametric empirical estimator;

Model fitting

Broadly three categories of ML model:

1. Full probabilistic model;

   → use your statistical prowess!

2. Parametric family without explicit probabilistic structure;

   → Empirical Risk Minimisation (ERM)

3. Local method constructing non-parametric empirical estimator;

   → empirical estimate of Bayes predictor

Model fitting: ERM

$\mathcal{F}$ a model family (or hypothesis space) parameterised by $\theta \in \Theta$.

$$\mathcal{F} = \{ f(\cdot \mid \theta) : \theta \in \Theta \}$$

Model fitting: ERM

$\mathcal{F}$ a model family (or hypothesis space) parameterised by $\theta \in \Theta$.

$$\mathcal{F} = \{ f(\cdot \,|\, \theta) : \theta \in \Theta \}$$

Assume dataset $\mathcal{D} = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)) \overset{iid}{\sim} \pi_{XY}^n$.

Fit a model $\hat{f} \in \mathcal{F}$ to $\mathcal{D}$ using *empirical risk minimisation* of a loss function $\mathcal{L}(\cdot, \cdot)$ as

$$\hat{f}(\cdot) = f(\cdot \,|\, \hat{\theta}) \text{ where } \hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(y, f(x \,|\, \theta))$$

Model fitting: local methods

Imagining target loss of interest is squared loss, we know optimal Bayes predictor is:

$$g^\star(\mathbf{x}) = \mathbb{E}[Y \mid X = \mathbf{x}]$$

Local methods just estimate this value directly using data “local” (under some metric) to $\mathbf{x}$.

Model fitting: local methods

Imagining target loss of interest is squared loss, we know optimal Bayes predictor is:

$$g^\star(\mathbf{x}) = \mathbb{E}[Y \mid X = \mathbf{x}]$$

Local methods just estimate this value directly using data “local” (under some metric) to $\mathbf{x}$.

e.g. k-nearest neighbour empirically estimates $\mathbb{E}[Y \mid X = \mathbf{x}]$ by selecting the $k \in \mathbb{N}^+$ observations in $\mathcal{D}$ which are ‘nearest’ (by some metric, $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$) to $\mathbf{x}$.

Model fitting: local methods

Imagining target loss of interest is squared loss, we know optimal Bayes predictor is:

$$g^\star(\mathbf{x}) = \mathbb{E}[Y \mid X = \mathbf{x}]$$

Local methods just estimate this value directly using data "local" (under some metric) to $\mathbf{x}$.

e.g. k-nearest neighbour empirically estimates $\mathbb{E}[Y \mid X = \mathbf{x}]$ by selecting the $k \in \mathbb{N}^+$ observations in $\mathcal{D}$ which are 'nearest' (by some metric, $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$) to $\mathbf{x}$.

- conceptually simple
- easy to implement
- hard to scale with dimensionality
- easy to overfit

Model fitting: statistical methods

Do **not** blindly perform ERM if $\exists$ plausible probabilistic model!

Unsurprisingly, *if* assumptions of full probabilistic model are approximately satisfied, then full likelihood or Bayesian methods usually give better model fit and better predictive performance.

Model fitting: statistical methods

Do **not** blindly perform ERM if $\exists$ plausible probabilistic model!

Unsurprisingly, *if* assumptions of full probabilistic model are approximately satisfied, then full likelihood or Bayesian methods usually give better model fit and better predictive performance.

Trivial Example: $Y \sim \mathrm{N}(\mu, \sigma^2)$

- Variance of mean: $\frac{\sigma^2}{n}$
- Variance of median: $\frac{\pi\sigma^2}{2n}$

Model fitting: statistical methods

Do **not** blindly perform ERM if $\exists$ plausible probabilistic model!

Unsurprisingly, *if* assumptions of full probabilistic model are approximately satisfied, then full likelihood or Bayesian methods usually give better model fit and better predictive performance.

Trivial Example: $Y \sim \mathrm{N}(\mu, \sigma^2)$

- Variance of mean: $\frac{\sigma^2}{n}$
- Variance of median: $\frac{\pi\sigma^2}{2n}$

$\implies$ favour computing mean (min sq loss) versus computing median (min abs loss) for either sq or abs loss!

Regularisation

Note standard approaches to regularisation (see APTS High-dim stats module) apply to machine learning too:

$$\arg\min_{\theta \in \Theta} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}(y, f(x \mid \theta)) + \lambda C(\theta)$$

where $C(\theta)$ measures model complexity; λ controls extent of regularisation.

eg, ridge regression, $C(\theta) = \|\theta\|_2^2$; or lasso regression, $C(\theta) = \|\theta\|_1$

Proper scoring rules (I)

Not all losses created equally!

As statisticians, we should care if whole probabilistic forecast is good, not just point estimate. Losses which are proper scoring rules (Gneiting and Raftery, 2007) ensure *calibration*.

Proper scoring rules (I)

Not all losses created equally!

As statisticians, we should care if whole probabilistic forecast is good, not just point estimate. Losses which are proper scoring rules (Gneiting and Raftery, 2007) ensure *calibration*.

Let $\mathcal{P}$ be the space of probability distributions on $\mathcal{Y}$. *Scoring rule* is function

$$S : \mathcal{P} \times \mathcal{Y} \to \mathbb{R}$$

giving numerical value to probabilistic prediction $P \in \mathcal{P}$ and associated outcome $y \in \mathcal{Y}$.

Proper scoring rules (I)

Not all losses created equally!

As statisticians, we should care if whole probabilistic forecast is good, not just point estimate. Losses which are proper scoring rules (Gneiting and Raftery, 2007) ensure *calibration*.

Let $\mathcal{P}$ be the space of probability distributions on $\mathcal{Y}$. *Scoring rule* is function

$$S : \mathcal{P} \times \mathcal{Y} \to \mathbb{R}$$

giving numerical value to probabilistic prediction $P \in \mathcal{P}$ and associated outcome $y \in \mathcal{Y}$.

A scoring rule is said to be a *proper scoring rule* if

$$\mathbb{E}_{Y \mid X} S(\pi_{Y \mid X}, Y) \ge \mathbb{E}_{Y \mid X} S(P, Y) \; \forall \, P \in \mathcal{P}$$

The rule is said to be *strictly* proper when equality occurs if and only if $P \equiv \pi_{Y \mid X}$.

Proper scoring rules (II)

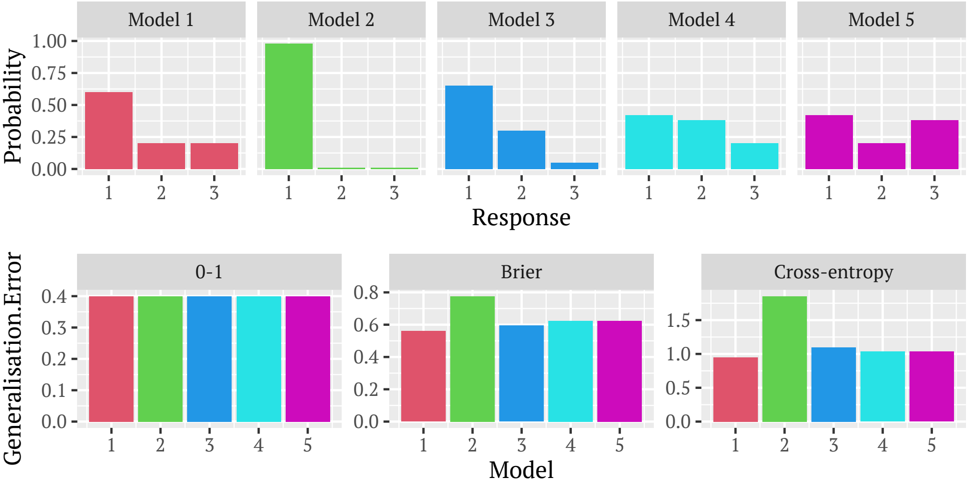
Regression:

- square loss: proper
- absolute loss: proper
- likelihood: strictly proper

Classification:

- 0-1 loss: proper
- cross entropy: strictly proper
- Brier: strictly proper

Proper scoring rules: example (based on Štrumbelj, 2018)



Limitations

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."

— Box and Draper (1987), pp.74

"[...] all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind."

— Box and Draper (1987), pp.424

Limitations

> *"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."*
>
> — *Box and Draper (1987), pp.74*

> *"[...] all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind."*
>
> — *Box and Draper (1987), pp.424*

- fewer assumptions does **not** mean $\exists$ universally best method;
- universal consistency is **not** get out of jail free: no such thing as infinite data!
- small sample size settings can benefit from simpler models and more assumptions.

Limitations

> *"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."*
>
> *— Box and Draper (1987), pp.74*
>
> *"[...] all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind."*
>
> *— Box and Draper (1987), pp.424*

- fewer assumptions does **not** mean $\exists$ universally best method;
- universal consistency is **not** get out of jail free: no such thing as infinite data!
- small sample size settings can benefit from simpler models and more assumptions.

Wolpert (1996) "no free lunch theorems": for any learning method there exists a π_{XY} for which it will be arbitrarily bad.

Theorem 7.1, Devroye et al. (1996)

Let $\varepsilon > 0$ be an arbitrarily small real value. For any integer n and classification rule g_n, there exists a distribution π_{XY} (for $\mathcal{Y}$ binary) with Bayes error zero, $\mathcal{E}^\star = 0$, such that

$$\bar{\mathcal{E}}_n = \mathbb{E}_{D_n} \left[\mathbb{E}_{XY} \left[\mathcal{L}(Y, g_n(X \mid D_n)) \right] \right] \geq \frac{1}{2} - \varepsilon$$

when $\mathcal{L}$ is 0-1 loss.

That is, for any sample size n there exists a distribution π_{XY} for which the learning method performs arbitrarily badly.

References I

Box, G.E.P., Draper, N.R. (1987). Empirical model-building and response surfaces, 1st ed, Wiley series in probability and statistics. Wiley. ISBN: 978-0471810339

Devroye, L., Györfi, L., Lugosi, G. (1996). A Probabilistic Theory of Pattern Recognition, 1st ed. Springer. ISBN: 978-0387946184

Gneiting, T., Raftery, A.E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102**(477), 359–378. DOI: 10.1198/016214506000001437

Štrumbelj, E. (2018). Predictive model evaluation. *Course Notes*. URL `https://file.biolab.si/textbooks/ml1/model-evaluation.pdf`

Vapnik, V.N. (1998). The Nature of Statistical Learning Theory, 2nd ed. Springer. ISBN: 978-0387987804

References II

Wolpert, D.H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation* **8**, 1341–1390. DOI: 10.1162/neco.1996.8.7.1341