COMS30035, Machine learning:
Key ML concepts

Edwin Simpson (based on slides by Rui Ponte Costa)

Department of Computer Science, SCEEM
University of Bristol

September 24, 2023

Textbooks

We will go over ML concepts following Chapter 1 of both textbooks:

- Bishop, C. M., Pattern recognition and machine learning (2006). Available for free here.
- Murphy, K., Machine learning a probabilistic perspective (2012). The book is also freely available here.

Agenda

- The different forms of machine learning:
  - Unsupervised learning
  - Supervised learning
  - Reinforcement learning
- Other important concepts in ML:
  - Overfitting
  - Model selection
  - The curse of dimentionality
  - No free lunch theorem
  - Parametric vs non-parametric models

The different forms of machine learning

- ML attempts to learn **models** of the world
  - Usually with many simplifications
  - Models are a way of understanding how **input** data relates to the **outputs**
  - E.g., a function that maps weather observations to predictions

The different forms of machine learning

- ML attempts to **learn** models of the world

The different forms of machine learning

- ML attempts to **learn** models of the world
  - Many different flavours of data are available!

The different forms of machine learning

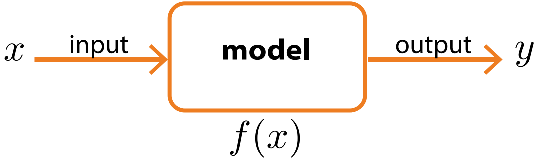
- ML attempts to **learn** models of the world
  - Many different flavours of data are available!
- The data available defines which form of learning we can use

The different forms of machine learning

- ML attempts to **learn** models of the world
  - Many different flavours of data are available!
- The data available defines which form of learning we can use
- However, the principle is always the same: model the data..
  - ..the model assumptions and data structure vary

Unsupervised learning

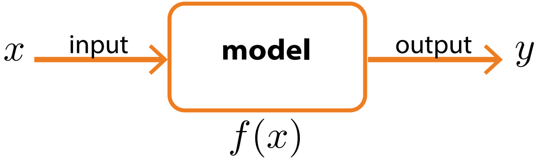
Only the input data is provided and models learn to extract patterns from the data.



[1] Note that virtually all models that do not use explicit teaching signals, such as targets/labels or rewards are unsupervised.

Unsupervised learning

Only the input data is provided and models learn to extract patterns from the data.



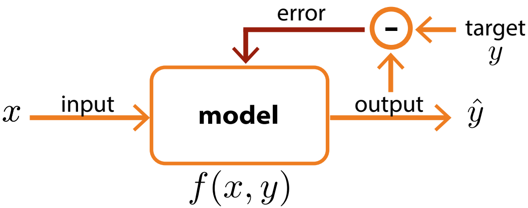
Example tasks [1]:

- Clustering: grouping similar items together (e.g., K-means, unsupervised HMM)
- Dimensionality reduction (PCA, ICA): finding a simplified representation of input data with fewer dimensions
- Density estimation (mixture models, language models): used to estimate the probabilities of input data points

[1] Note that virtually all models that do not use explicit teaching signals, such as targets/labels or rewards are unsupervised.

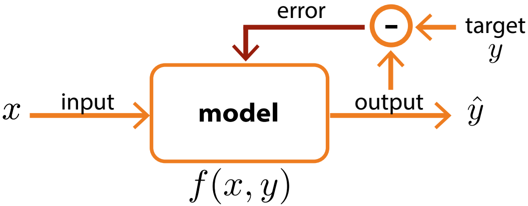
Supervised learning

Each input is paired with an output – a “label” or “target” – and the model is trained to minimise the error (difference) between its output and the target.



Supervised learning

Each input is paired with an output – a “label” or “target” – and the model is trained to minimise the error (difference) between its output and the target.

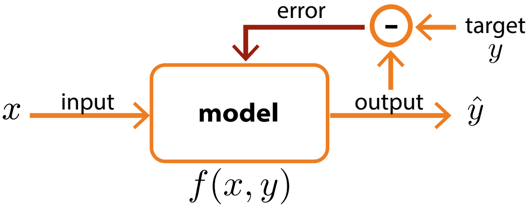


Example tasks:

- Regression: numerical outputs (e.g., air temperatures over time)
- Classification: category labels (e.g. , dog or a bagel?)

Supervised learning

Each input is paired with an output – a “label” or “target” – and the model is trained to minimise the error (difference) between its output and the target.



Example tasks:

- Regression: numerical outputs (e.g., air temperatures over time)
- Classification: category labels (e.g. , dog or a bagel?)

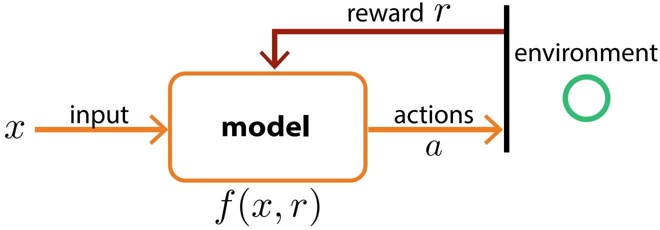
Many learning methods can be used for both regression and classification:

- Supervised neural networks
- Support vector machines
- Decision trees

Reinforcement learning

The correct outputs are not given, but the model receives a reward (or punishment) depending on its actions.

RL deals with dynamic environments where agents carry out sequences of actions. It is inspired by animal behaviour. RL is seen as a field on its own – *we do **not** teach RL on this unit*.

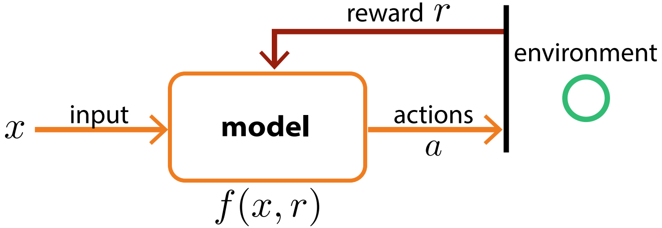


[2] If you would like to learn more see Information Processing and the Brain in your 4th year.

Reinforcement learning

The correct outputs are not given, but the model receives a reward (or punishment) depending on its actions.

RL deals with dynamic environments where agents carry out sequences of actions. It is inspired by animal behaviour. RL is seen as a field on its own – *we do **not** teach RL on this unit*.



Examples [2]:

- Temporal difference learning
- Deep reinforcement learning (uses neural networks)

[2] If you would like to learn more see Information Processing and the Brain in your 4th year.

Underfitting vs overfitting

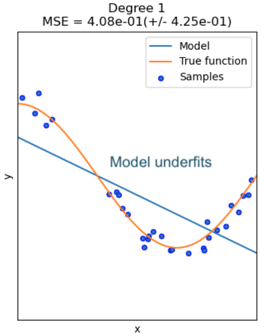
Underfitting: A model that is too simple – it should be as "simple as possible, but no simpler."[3]

[3] As Einstein used to say when formulating theories "Everything should be made as simple as possible, but no simpler."

Underfitting vs overfitting

Underfitting: A model that is too simple – it should be as "simple as possible, but no simpler."[3]

Example from *sk-learn* (click here):

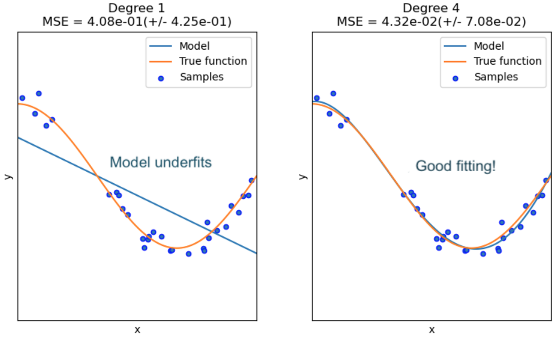


[3] As Einstein used to say when formulating theories "Everything should be made as simple as possible, but no simpler."

Underfitting vs overfitting

Underfitting: A model that is too simple – it should be as "simple as possible, but no simpler."[3]

Example from *sk-learn* (click here):



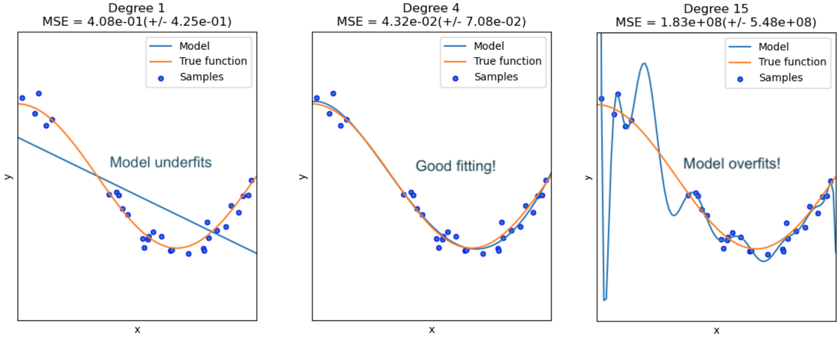
[3] As Einstein used to say when formulating theories "Everything should be made as simple as possible, but no simpler."

Underfitting vs overfitting

Underfitting: A model that is too simple – it should be as "simple as possible, but no simpler."[3]

Overfitting: A model that fits minor variations or noise; highly flexible models are particularly prone to overfitting.

Example from *sk-learn* (click here):

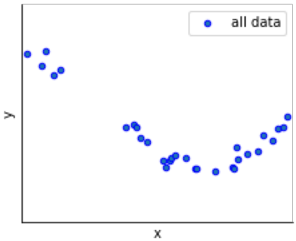


[3] As Einstein used to say when formulating theories "Everything should be made as simple as possible, but no simpler."

Model selection

There is an infinite number of models, how do we choose just one?
Answer: We perform model selection to reduce under/overfitting. [4].

A common method is to *split the dataset*. Lets look again at the data used in the previous slide

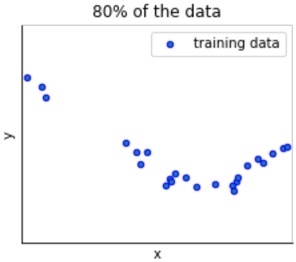


[4]Note that models that overfit or underfit fail to **generalise** to new data, this idea underlies model selection.

Model selection

Lets split the full dataset into:

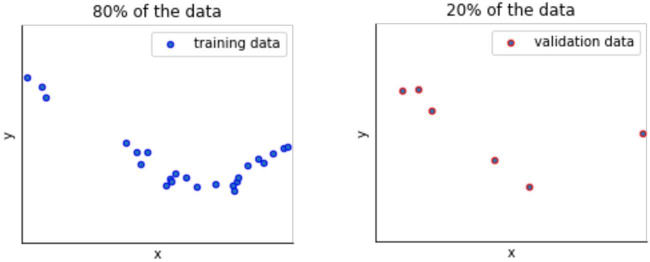
- **Training dataset**: Used for training/optimising your model (e.g. use 80% of the full dataset)



Model selection

Lets split the full dataset into:

- **Training dataset**: Used for training/optimising your model (e.g. use 80% of the full dataset)
- **Validation dataset**: Used *only* for validating your model (e.g. use 20% of the full dataset)



Model selection

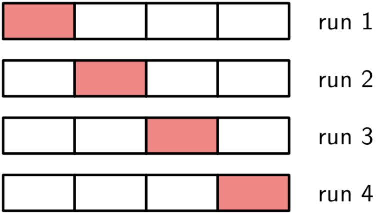
Relying only on the validation dataset to select our models can lead us to overfit to that data, in particular for small datasets and iterative methods. So it is often common to use a third subset, the *testing dataset*.

- **Testing dataset**: Used to test the model for general fitting quality after the optimisation procedure has finished (e.g. use 10% of the full dataset).

Model selection

However, simply splitting the data means that we end up with less data for training the model. A solution is to cycle over multiple subsets of the data using *cross-validation*.

- **Cross-validation**: The original data is split into S groups so that $(S-1)/S$ data is used for training. It is common to set S to a relatively low number, e.g.: $S = 4$, which gives 4-fold cross validation using 3 (75% of the data) subsets for training (white blocks) and 1 for validation (red block) for each run [5]:



[5] If $S = N$ where N is the full number of data samples it gives the *leave-one-out* method.

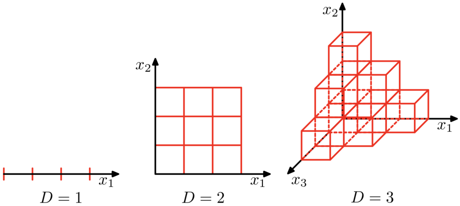
Agenda

- The different forms of machine learning:
  - Unsupervised learning
  - Supervised learning
  - Reinforcement learning
- Other important concepts in ML:
  - Overfitting
  - Model selection
  - **The curse of dimentionality**
  - **No free lunch theorem**
  - **Parametric vs non-parametric models**

Curse of dimentionality in ML

1D and 2D spaces can be covered by data easily, but for higher dimensions this is no longer feasible.

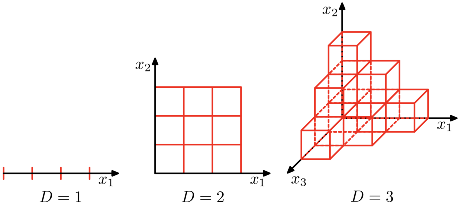
- If we were to divide the space into cells we would quickly need an exponentially large quantity of data to fill in all cells (see schematic below).



Curse of dimentionality in ML

1D and 2D spaces can be covered by data easily, but for higher dimensions this is no longer feasible.

- If we were to divide the space into cells we would quickly need an exponentially large quantity of data to fill in all cells (see schematic below).
- However, its often possible to find effective algorithms for two reasons (Bishop book):
  - Data is often restricted to specific regions of the much bigger spaces – i.e. effective dimentionality is much smaller.
  - Data typically has smoothness properties – i.e. small changes in the input variables will lead to small changes in the output variables.



No free lunch theorem

All models are wrong, but some models are useful. — George Box 1976

No free lunch theorem

All models are wrong, but some models are useful. — George Box 1976

- Using model selection we can obtain a *good model*.
- *But* there is no universally best model – **no free lunch theorem** (Wolpert 1996).

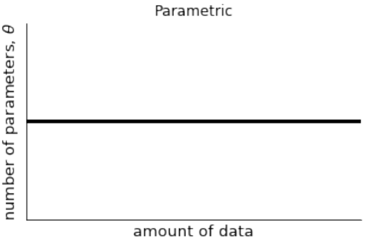
No free lunch theorem

All models are wrong, but some models are useful. — George Box 1976

- Using model selection we can obtain a *good model*.
- *But* there is no universally best model – **no free lunch theorem** (Wolpert 1996).
- Why? Models always make assumptions and these often do not generalise across domains – different domains need different models.

Parametric vs non-parametric models

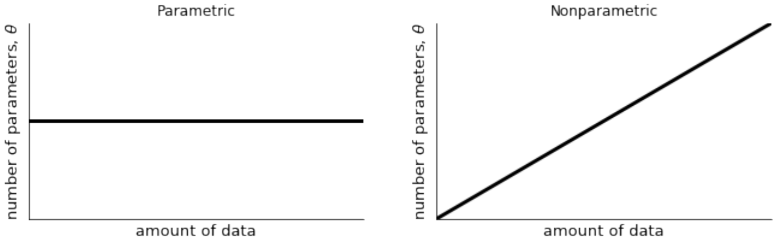
- **Parametric**: Model assumes fixed number parameters θ.



[6]Most nonparametric models are hybrid models with parametric (non-flexible) components.

Parametric vs non-parametric models

- **Parametric**: Model assumes fixed number parameters θ.
- **Non-parametric**: Model parameters θ grows with the amount of data.[6]



[6] Most nonparametric models are hybrid models with parametric (non-flexible) components.

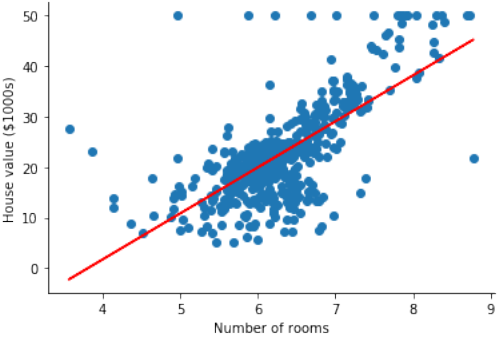
Parametric models

- **Pros**: Simpler, fast to fit and require less data.
- **Cons**: Limited and better for simpler problems/datasets.

Parametric models

- **Pros**: Simpler, fast to fit and require less data.
- **Cons**: Limited and better for simpler problems/datasets.

Example: Linear regression model $y = ax + b$ assumes 2 parameters $\theta = \{a, b\}$, where a is the slope and b the y-intercept.



Non-parametric models

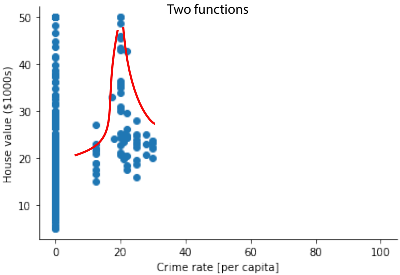
- **Pros**: Flexible (i.e. can infer which functions to use), weak assumptions, can give better models.
- **Cons**: Need more data, slower to train (more parameters), risk of overfitting.

[7] Note that this is an hypothetical algorithm to illustrate the increase in number of parameters as a function of data.

Non-parametric models

- **Pros**: Flexible (i.e. can infer which functions to use), weak assumptions, can give better models.
- **Cons**: Need more data, slower to train (more parameters), risk of overfitting.

Example: Nonparametric regression with an algorithm[7] that automatically detects which polynomial functions to use.

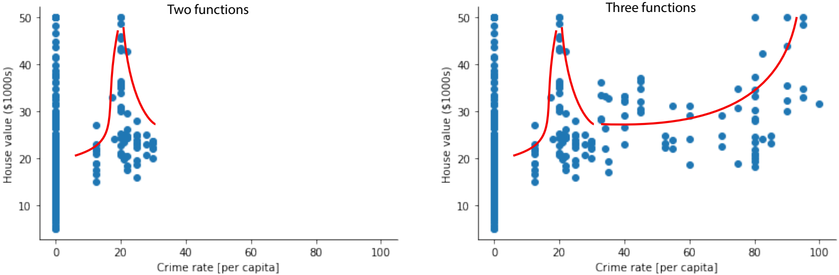


[7] Note that this is an hypothetical algorithm to illustrate the increase in number of parameters as a function of data.

Non-parametric models

- **Pros**: Flexible (i.e. can infer which functions to use), weak assumptions, can give better models.
- **Cons**: Need more data, slower to train (more parameters), risk of overfitting.

Example: Nonparametric regression with an algorithm[7] that automatically detects which polynomial functions to use.



[7] Note that this is an hypothetical algorithm to illustrate the increase in number of parameters as a function of data.

Questions

- Please also post questions on Teams or raise in the labs or next lecture.

Questions

- Please also post questions on Teams or raise in the labs or next lecture.
- Lab 1: Find it on Blackboard and try to set up your Python environment with the required packages.

Questions

- Please also post questions on Teams or raise in the labs or next lecture.

- Lab 1: Find it on Blackboard and try to set up your Python environment with the required packages.
- Quiz time! Go to Blackboard unit page » Quizzes » Lecture 1. Should take you less than 3 minutes