Uncertainty in Modeling

Dependable Systems

Dependable Systems can be found in many forms and application domains, but especially in transportation systems, medical systems and recently in the domain of IoT and Industry 4.0.*

*Industry 4.0 has been defined as “a **name for the current trend of automation and data exchange in manufacturing technologies**, including cyber-physical systems, the Internet of things, cloud computing and cognitive computing and creating the smart factory”

Dependable AI

Accuracy : The prediction accuracy is the basic ability of a trustworthy model. Trustworthy NNs are expected to generate accurate output, consistent with the ground truth, as much as possible;

Reliability: Trustworthy NNs should be resilient and secure. In other words, they must be robust against different potential threats, such as inherent noise, distribution shift, and adversarial attacks;

Explainability: The model itself must be explainable for the prediction, which can help humans to enhance understanding, make decisions and take further actions;

Privacy protection: Trustworthy NNs are required to ensure full privacy of the models as well as data privacy.

Pillars of Security

- **Confidentiality** is satisfied if data or objects are not read by an unauthorized party.
- **Integrity** is satisfied if data or objects are not changed (written) or generated by an unauthorized party.
- **Authenticity** is satisfied if an author of data or an object is who it claims to be.
- **Availability** is satisfied if data, objects, or services are available.

Can AI-based Components be Part of Dependable Systems?

Machine learning seeks to **develop methods for computers to improve their performance at certain tasks** on the basis of observed data.

Typical example:

- detecting pedestrians in images taken from an autonomous vehicle,
- classifying gene-expression patterns from leukaemia patients into subtypes by clinical outcome,
- or translating English sentences into French.
- scope of machine-learning tasks is even broader than these pattern classification or mapping tasks, and can include optimization and decision making, compressing data and automatically extracting interpretable models from data.

Data: The Treasure

- Almost all machine-learning tasks can be formulated as making inferences about missing or latent data from the observed data [inference, prediction or forecasting]

  Example :

  - Consider classifying people with leukaemia into one of the four main subtypes of this disease on the basis of each person's measured gene-expression patterns. Observed data : pairs of gene-expression patterns and labelled subtypes, Unobserved or missing data: subtypes for new patients.
    - **To make inferences about unobserved data from the observed data:**
      - **the learning system needs to make some assumptions**
      - **taken together these assumptions constitute a model.**

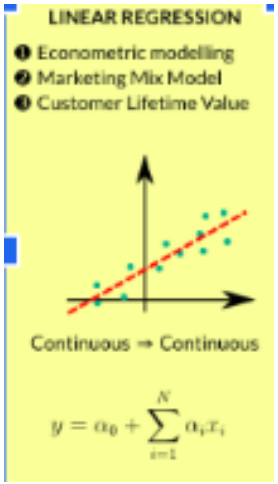
Model

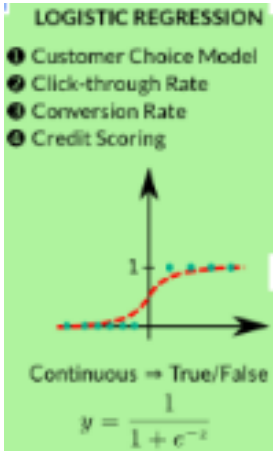
- A model can be very simple and rigid [classic statistical linear regression model]
- Model can be complex and flexible [llarge and deep neural network]
- A model is considered to be well defined if it can make forecasts or predictions about unobserved data having been trained on observed data
    - if the model cannot make predictions it cannot be falsified, in the sense of the philosopher Karl Popper's proposal for evaluating hypotheses, or as the theoretical physicist Wolfgang Pauli said the model is "not even wrong").

Any sensible model will be uncertain when predicting unobserved data, uncertainty plays a fundamental part in modelling.

Choice of Models

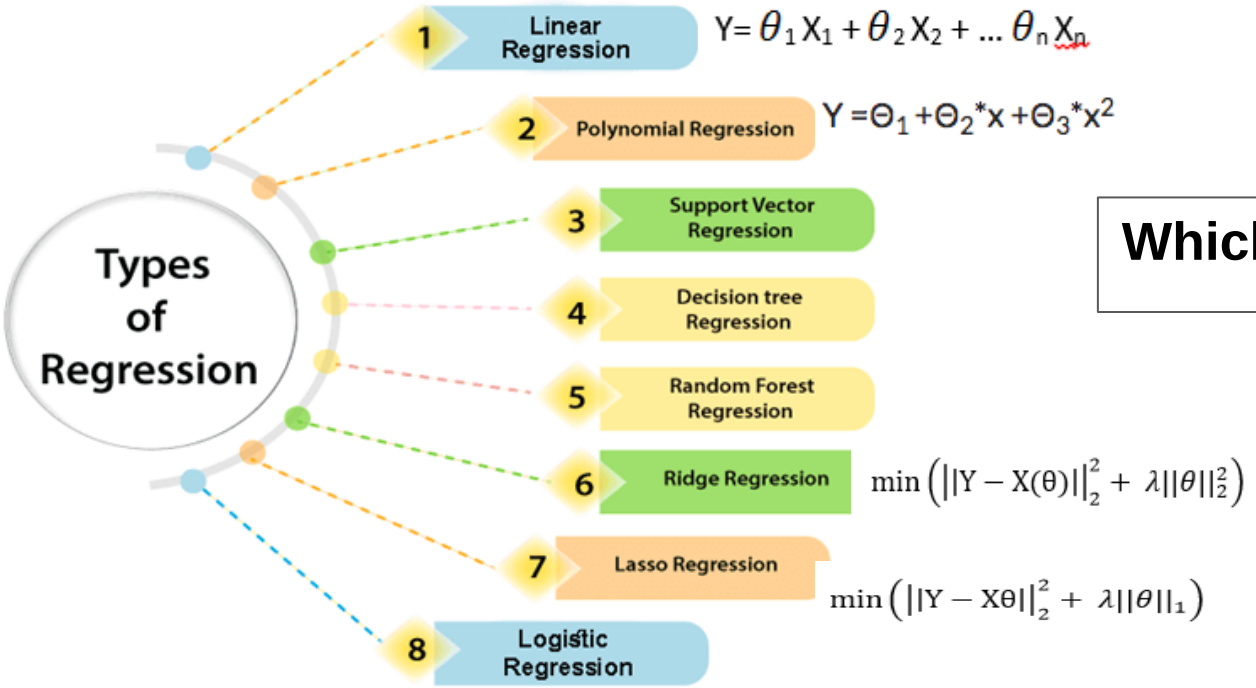
Rigid Linear Regression





- Linear Regression is greatly affected by the presence of Outliers, linearity assumption
  - An outlier is an unusual observation of response y, for some given predictor x.
- Logistic regression assumes **linearity of independent variables and log odds**.

Regression analysis helps in the prediction of a continuous variable

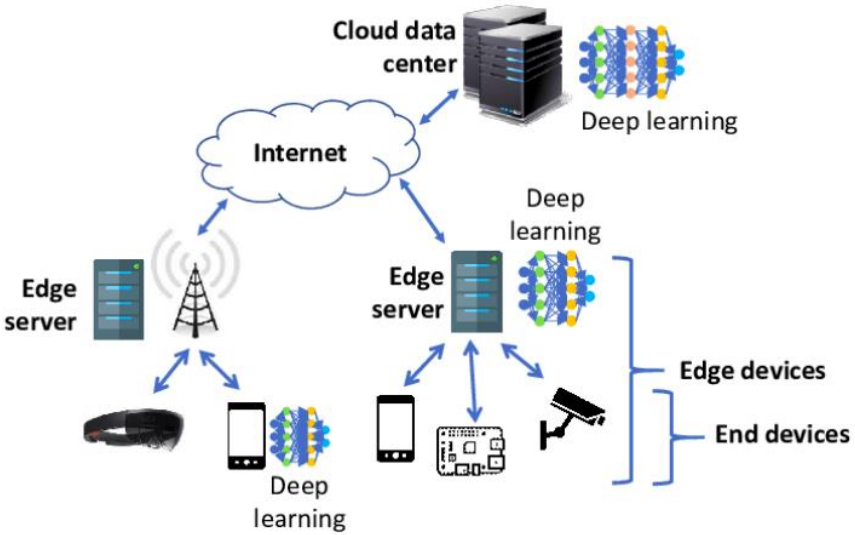


Which model to choose?

Flexible deep NN

DNNs is a challenge since they typically require billions of expensive arithmetic computations.

- DNNs are typically deployed in ensemble to boost accuracy performance, which further exacerbates the system requirements.
- This computational overhead is an issue for many platforms, e.g. data centers and embedded systems, with tight latency and energy budgets.
- Flexible DNNs: achieves large reduction in average inference latency while incurring small to negligible accuracy drop



Deep learning can execute on edge devices (i.e., end devices and edge servers) and on cloud data centers.

Problem 1: Uncertainty in Modelling

Uncertainty in Modelling

- At the lowest level, model uncertainty is introduced from **measurement noise**, for example, pixel noise or blur in images.
- At higher levels, a model may have many parameters, such as the coefficients of a linear regression, and there is **uncertainty about which values of these parameters will be good at predicting new data**.
- Finally, at the highest levels : uncertainty about even the general **structure of the model:**
    - Is linear regression or a neural network appropriate, if the latter, how many layers should it have, and so on.

Probabilistic approach to modelling

- uses probability theory to express all forms of uncertainty .
- probability distributions are used to represent all the uncertain unobserved quantities in a model
    - structural, parametric and noise-related
    - how they relate to the data.

Probabilistic approach to modelling

- basic rules of probability theory are used to infer the unobserved quantities given the observed data.
- Learning from data occurs through the transformation of the prior probability distributions (defined before observing the data), into posterior distributions (after observing data).
- **The application of probability theory to learning from data is called Bayesian learning**

Bayesian machine learning: How to choose ?

There are two simple rules that underlie probability theory: the sum rule:

$$P(x) = \sum_{y \in Y} P(x,y)$$

and the product rule:

$$P(x,y) = P(x)P(y \mid x).$$

- x and y: observed or uncertain quantities, taking values in some sets X and Y, respectively.
- For example, x and y might relate to the weather in Cambridge and London, respectively, both taking values in the set X=Y={rainy,cloudy,sunny}
- P(x) : probability of x
- P(x,y) is the joint probability of observing x and y,
- P(y|x) is the probability of y conditioned on observing the value of x

Bayesian machine learning: How to choose?*

- Since P(x,y) and P(y,x) are commutative: $$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)} = \frac{P(x \mid y)P(y)}{\sum_{y \in Y} P(x,y)}$$

- Apply probability theory to machine learning by replacing the symbols above:
  - Replace x by D to denote the observed data
  - Replace y by θ to denote the unknown parameters of a model

$$P(\theta \mid D, m) = \frac{P(D \mid \theta, m)P(\theta \mid m)}{P(D \mid m)}$$

where $P(D \mid \theta, m)$ is the likelihood of parameters θ in model m, $P(\theta \mid m)$ is the prior probability of θ and $P(\theta \mid D, m)$ is the posterior of θ given data D.

Ref: *https://www.nature.com/articles/nature14541.pdf

Compositional probabilistic models

- Simple probability distributions over single or a few variables can be composed to form the building blocks of larger, more complex models.
- Representing such compositional probabilistic models: graphical models

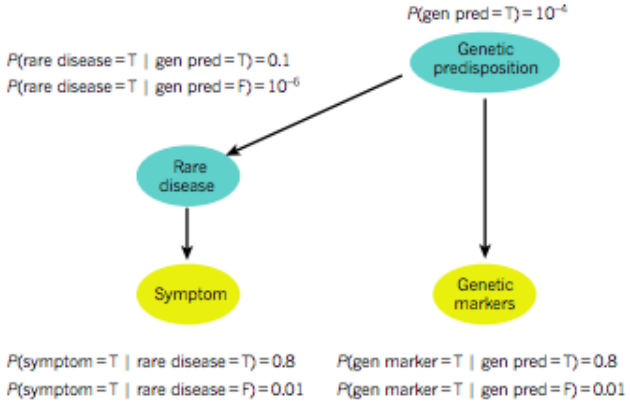


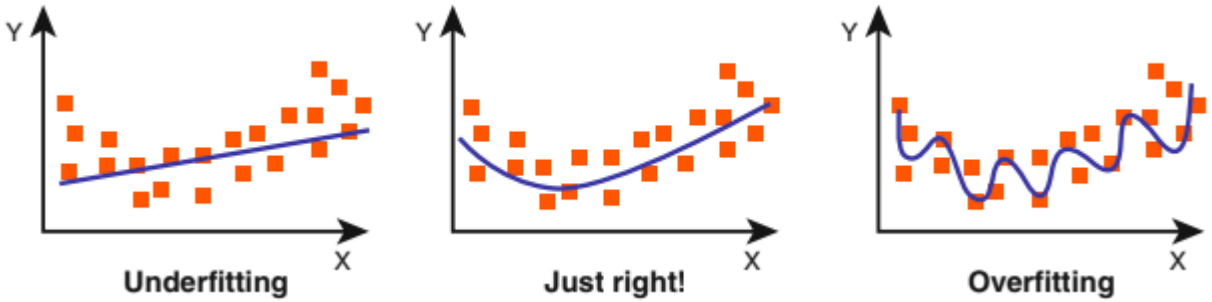
Figure 1 | Bayesian inference. A simple example of Bayesian inference applied to a medical diagnosis problem. Here the problem is diagnosing a rare disease using information from the patient's symptoms and, potentially, the patient's genetic marker measurements, which indicate predisposition (gen pred) to this disease. In this example, all variables are assumed to be binary. T, true; F, false. The relationships between variables are indicated by directed arrows and the probability of each variable given other variables they directly depend on is also shown. Yellow nodes denote measurable variables, whereas green nodes denote hidden variables. Using the sum rule (Box 1), the prior probability of the patient having the rare disease is: $P(\text{rare disease} = \text{T}) = P(\text{rare disease} = \text{T}|\text{gen pred} = \text{T})\,p(\text{gen pred} = \text{T}) + p(\text{rare disease} = \text{T}|\text{gen pred} = \text{F})\,p(\text{gen pred} = \text{F}) = 1.1 \times 10^{-5}$. Applying Bayes rule we find that for a patient observed to have the symptom, the probability of the rare disease is: $p(\text{rare disease} = \text{T}|\text{symptom} = \text{T}) = 8.8 \times 10^{-4}$, whereas for a patient observed to have the genetic marker (gen marker) it is $p(\text{rare disease} = \text{T}|\text{gen marker} = \text{T}) = 7.9 \times 10^{-4}$. Assuming that the patient has both the symptom and the genetic marker the probability of the rare disease increases to $p(\text{rare disease} = \text{T}|\text{symptom} = \text{T}, \text{gen marker} = \text{T}) = 0.06$. Here, we have shown fixed, known model parameters, that is, the numbers $\theta = (10^{-4}, 0.1, 10^{-6}, 0.8, 0.01, 0.8, 0.01)$. However, both these parameters and the structure of the model (the presence or absence of arrows and additional hidden variables) could be learned from a data set of patient records using the methods in Box 1.

Problem 2: Overfitting of ML nets

Why Overfitting?

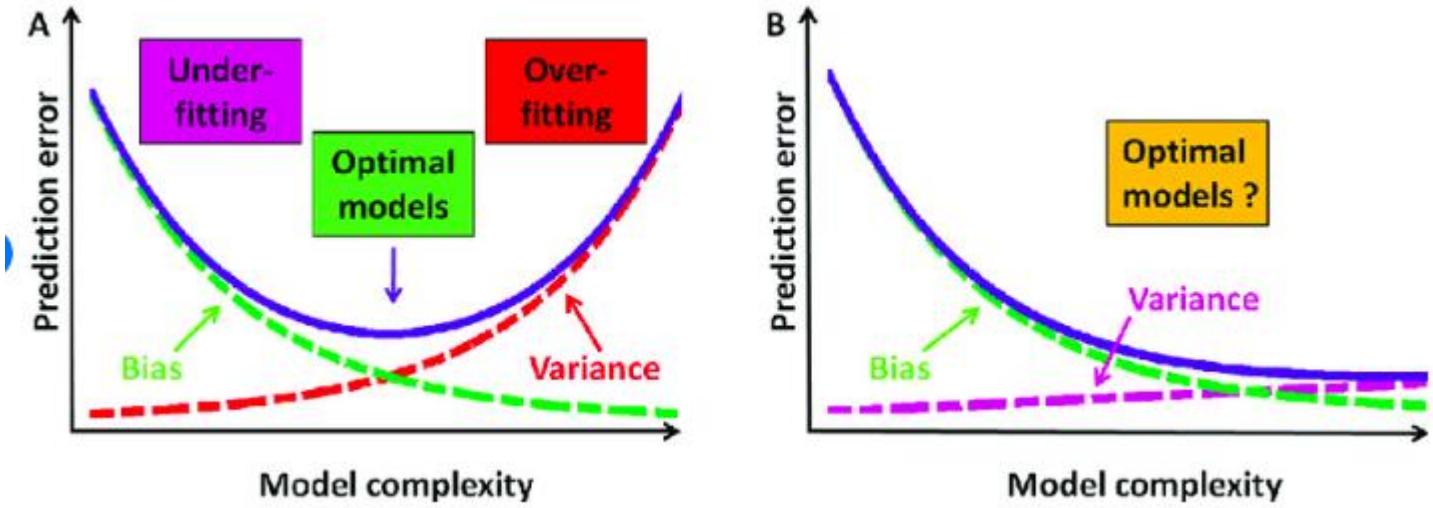
- Overfitting is **a modeling error in statistics** that occurs when a function is too closely aligned to a limited set of data points
- Example:
  - A traffic sign recognition system has to learn signs, but all STOP signs are only captured in an alley.
  - The ML algorithm is likely to learn, that a sign in an alley is a STOP sign and therefore detecting every traffic sign in an alley as such. Or a STOP sign, that is not located in an alley can not be detected.
  - This is caused by a **low variance in the data, with a high bias**

Overfitting underfitting details



A network is overfitting when a model's training error (computed on a training set) is much lower than its generalization error (computed on a test or validation set).

This is opposite to underfitting, when a model is not able to obtain a sufficiently low error value on its training set.



- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.
- Variance refers to the changes in the model when using different portions of the training data set [**variance** is the expectation of the squared deviation of a random variable from its population mean or sample mean.]

Problem 3: Edge Cases

Edge Cases

- **Unexpected scenarios that occur so irregularly or even seldom are called Edge Cases**
- often not covered in any training data, simply because they are not anticipated during the system development.
- All can possibly lead to accidents, when the systems deals with them in a wrong way.

Example:

- A vehicle that is only used in sunny weather for a long time in the same environment, will override some of the learned features trained by the manufacturer.
- handling intersections and performing turns as well as crossing the intersection.

Solution

- Learning algorithms could continue to **evolve while in operation**
- this approach is also called continuous or on-the-fly learning.
- Actually this approach is often seen as similar to human learning processes.
- Drawback:
  - potentially unwanted behavior
  - the training can not be influenced in a suitable way
  - dynamic changes to a certified system in general is not permitted by the current regulations and standards

Test Case

- To understand the implication of critical components based on machine learning, HAW Hamburg uses miniature vehicles

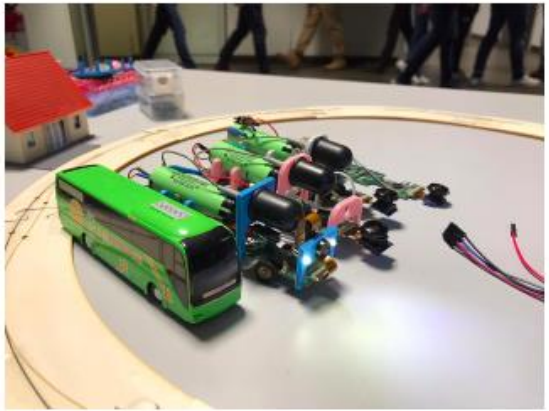


Fig. 3: Autonomous miniature vehicle type 2 (“truck”, large). The latter is designed to carry an FPGA board while the former is controlled by an ESP32 micro controller. (©Wunderland, 2019; Tiedemann, 2019, under CC-BY 4.0)

Testing on use cases:

- obstacle detection for autonomous driving.
- recognition of traffic signs
- obstacles in the driving lane
- route detection
- Fault tree analysis

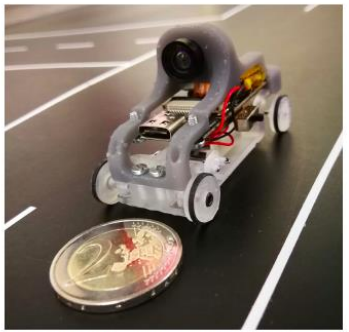


Fig. 2: Autonomous miniature vehicle type 1 (“sedan”, small) (©Tiedemann, 2019, under CC-BY 4.0)

In Summary

Consider *adversarial examples*: small perturbations of input examples that make even a highly accurate ML model give incorrect predictions.

- Adversarial examples can be used to regularize the training procedure and make a model robust to small perturbations of data (which is *a special case of stability*).
- Adversarial examples can be used as explanations by providing the minimal changes in the input that would alter the model prediction on it (*counterfactual explanations*).
- Adversarial examples that only change certain protected attributes like gender or race can be used to verify and optimize for fairness (*fairness audit*).

Refs

- Torge Hinrichs, Bettina Buth:**Can AI-based Components be Part of Dependable Systems?** IV 2020: 226-231, https://ieeexplore.ieee.org/document/9304740

- **Reliable Machine Learning,** https://www.microsoft.com/en-us/research/group/reliable-machine-learning/