



Introduction to Machine Learning

Uniq+ AI/ML Week 1 Training

Jin Xu, Leo Klarner

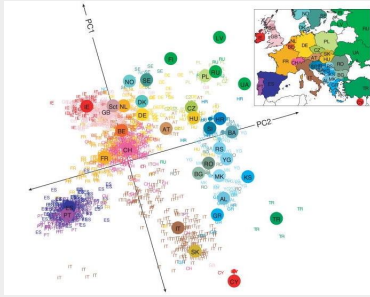
What is Machine Learning

“Field of study that gives computers the ability to learn without being explicitly programmed.”

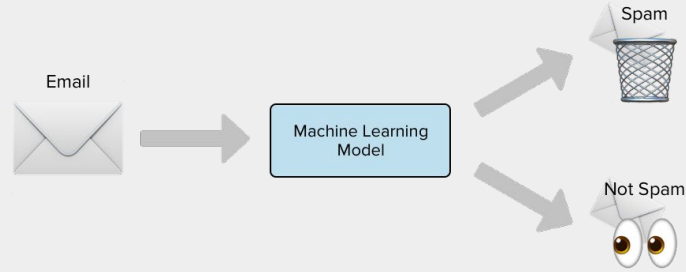
—————Arthur Samuel, 1959

“A computer program is said to learn from experience E with respect to some class of tasks T , and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”

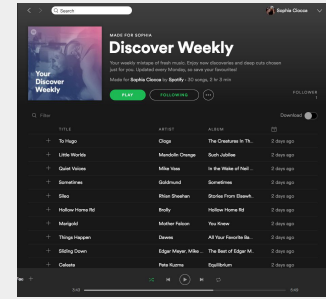
—————Tom Mitchell, 1997



Genetic Geography¹



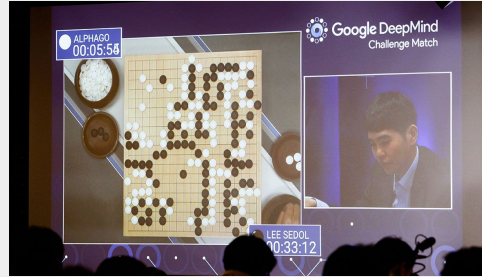
Spam Email Filter



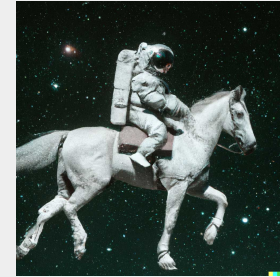
Music / Youtube Recommendation



Self Driving Cars



AlphaGo²



DALL·E 2³

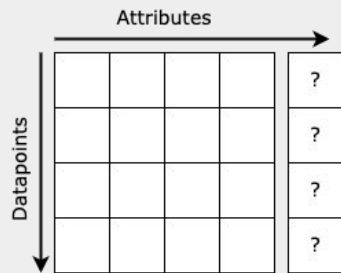
¹ Novembre, John, et al. "Genes mirror geography within Europe." *Nature* 456.7218 (2008): 98-101.

² <https://www.deepmind.com/research/highlighted-research/alphago>

³ <https://openai.com/dall-e-2/>

What is Machine Learning

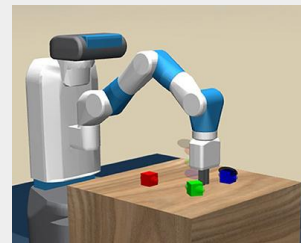
“Machine learning is all about **learning from data**, and there is generally a focus on making predictions at unseen data points.” ¹



Tabular data



Image datasets



Episodes of interactions between agents and environments



Audio data

¹ From lecture notes for SC4/SM8 Advanced Topics in Statistical Machine Learning, Tom Rainforth.

Supervised Learning

Learn with supervision (labels)

“Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.”

Unsupervised Learning

Learn patterns from unlabeled data

“Unsupervised learning is a type of algorithm that learns patterns from untagged data.”

——— From wikipedia

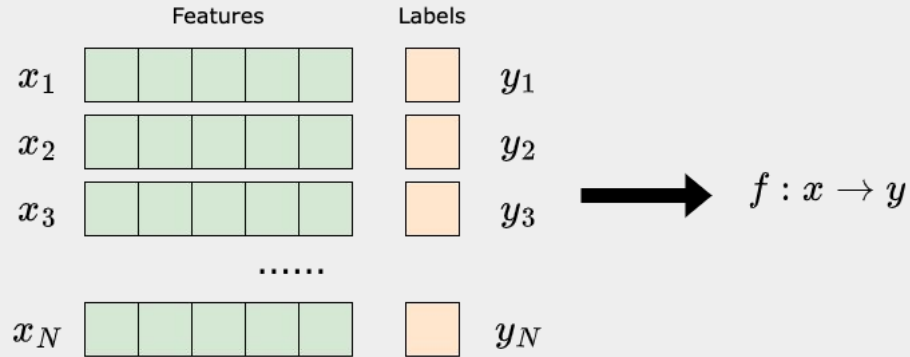
Supervised Learning



We have access to a training set consists of input-output pairs: $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$.

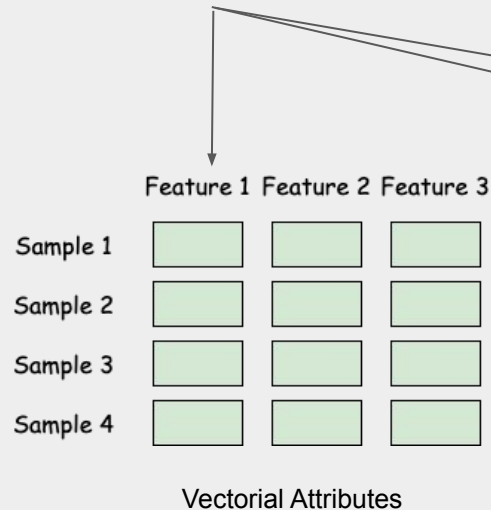
We hope to learn an predictive model f that takes an input x and predicts its corresponding output y .

Supervised Learning



We have access to a training set consists of input-output pairs: $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$.

We hope to learn an predictive model f that takes an input x and predicts its corresponding output y .

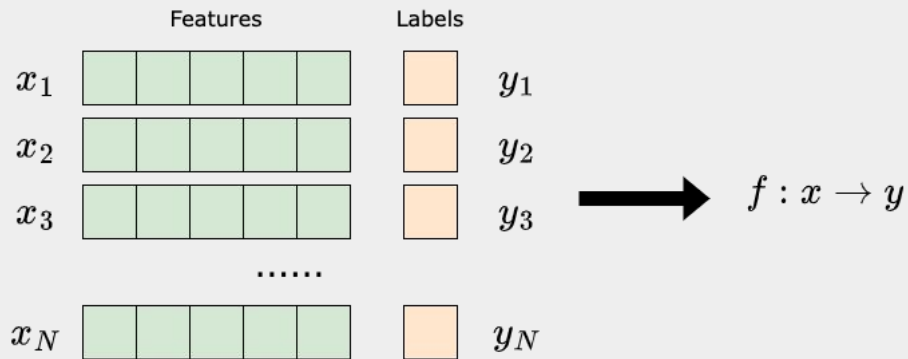


Images



Sequence

Supervised Learning



We have access to a training set consists of input-output pairs: $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$.

We hope to learn an predictive model f that takes an input x and predicts its corresponding output y .

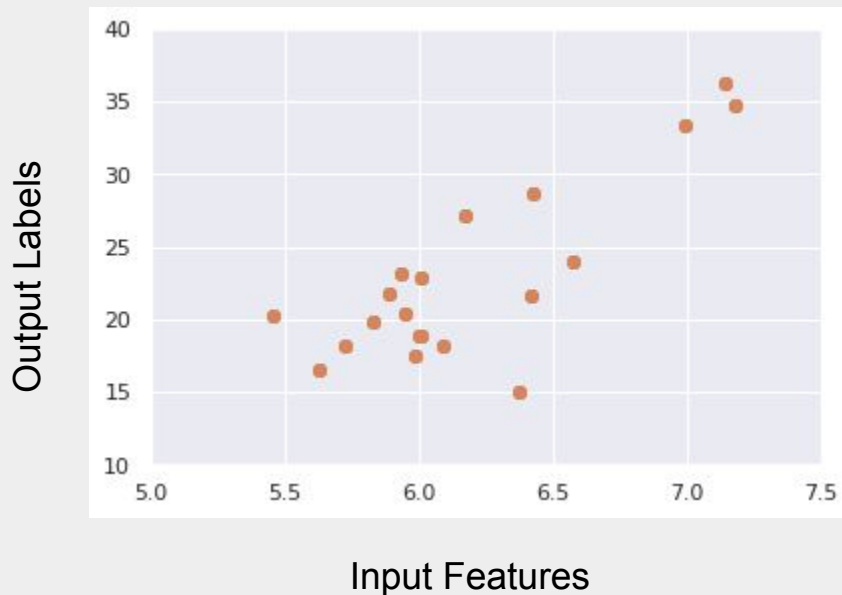
Real number
scalars as labels

Classes as labels

Regression

Classification

Supervised Learning: Regression



Given $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$

we find a model

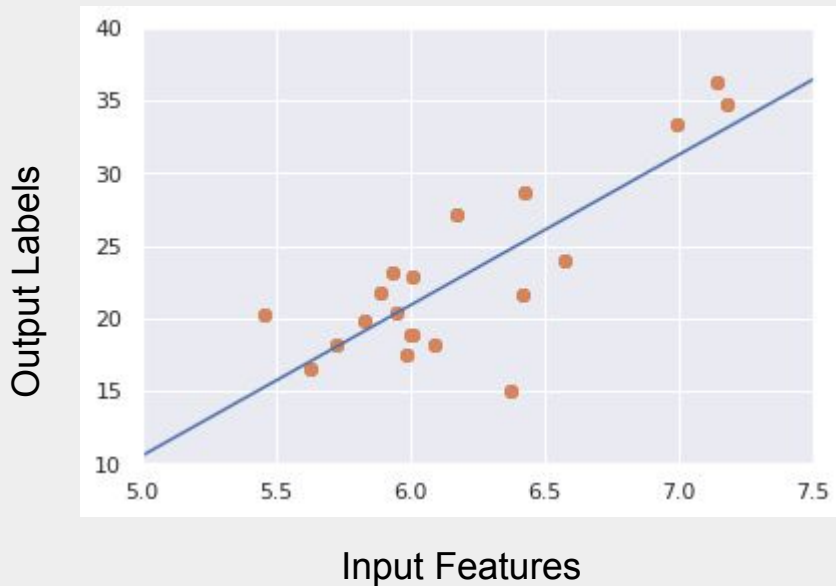
$$\hat{y} = f(x; \theta)$$

by solving

$$\arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N (f(x_n; \theta) - y_n)^2$$

(The squared loss is not the only possibility
but just a frequently used one!)

Supervised Learning: Linear Regression



Given $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$

we find a model

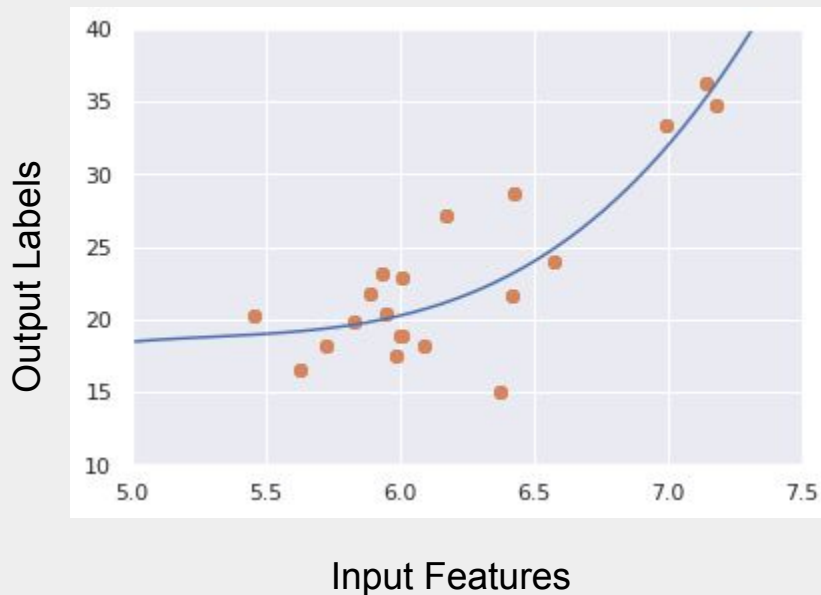
$$\hat{y} = f(x; \theta = \{w, b\}) = wx + b$$

by solving

$$\arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N (f(x_n; \theta) - y_n)^2$$

Supervised Learning: Polynomial Regression

Degree d=3 Polynomial Regression



Given $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$

we find a model

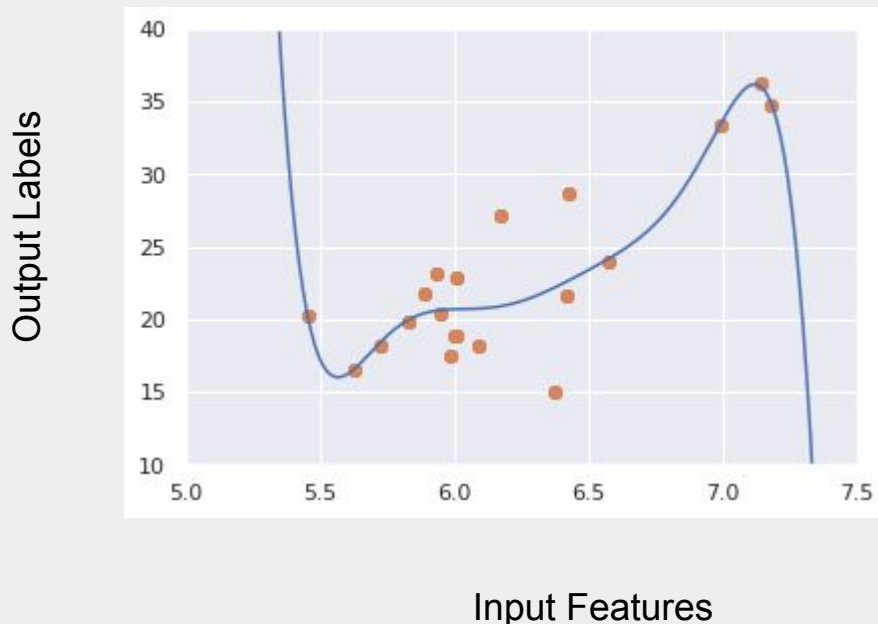
$$\hat{y} = f(x; \theta = \{w_0, \dots, w_d\}) = \sum_{i=1}^d w_i x^i + w_0$$

by solving

$$\arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N (f(x_n; \theta) - y_n)^2$$

Supervised Learning: Polynomial Regression

Degree d=10 Polynomial Regression



Given $D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$

we find a model

$$\hat{y} = f(x; \theta = \{w_0, \dots, w_d\}) = \sum_{i=1}^d w_i x^i + w_0$$

by solving

$$\arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N (f(x_n; \theta) - y_n)^2$$

Supervised Learning: Classification

Iris flower classification problem:

Attributes: Petal Length, Petal Width, Sepal Length, Sepal width.

Labels: Class(Species).

$y \in \{0, 1, 2\} ?$



Supervised Learning: Classification

Iris flower classification problem:

Attributes: Petal Length, Petal Width, Sepal Length, Sepal width.

Labels: Class(Species).



$$y \in \{0, 1, 2\} ?$$

Class labels should be unordered and exclusive

One-hot encoding

Iris Type	Versicolor	Setosa	Virginica
Versicolor	1	0	0
Setosa	0	1	0
Virginica	0	0	1

Supervised Learning: Learning a Classifier

It is easy to parameterise a model with unconstrained real-valued outputs:

$$\hat{y} = f(x; \theta), \hat{y} \in \mathbb{R}$$

How can we construct a binary classifier?

Supervised Learning: Learning a Classifier

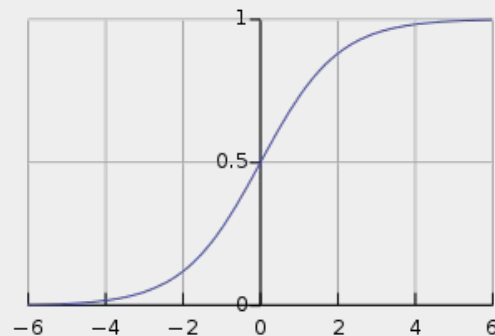
It is easy to parameterise a model with unconstrained real-valued outputs:

$$\hat{y} = f(x; \theta), \hat{y} \in \mathbb{R}$$

How can we construct a binary classifier?

$$p(y = 1|x; \theta) = \text{Sigmoid}(f(x; \theta))$$

$$\text{Sigmoid}(\phi) = \frac{1}{1 + \exp(-\phi)}$$



Supervised Learning: Learning a Classifier

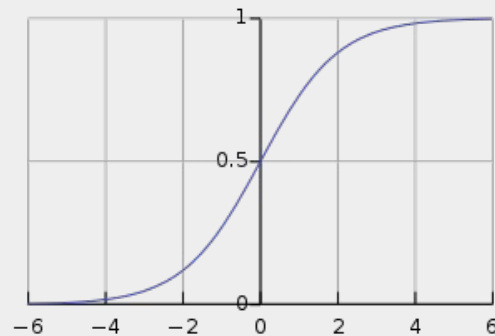
It is easy to parameterise a model with unconstrained real-valued outputs:

$$\hat{y} = f(x; \theta), \hat{y} \in \mathbb{R}$$

How can we construct a binary classifier?

$$p(y = 1|x; \theta) = \text{Sigmoid}(f(x; \theta))$$

$$\text{Sigmoid}(\phi) = \frac{1}{1 + \exp(-\phi)}$$



What if there are multiple (more than 2) classes?

$$p(y = j|x; \theta) = \frac{\exp(-f_j(x; \theta))}{\sum_i \exp(-f_i(x; \theta))}$$

where the function f outputs a C (number of classes) dimensional outputs.

Supervised Learning: Learning a Classifier

It is easy to parameterise a model with unconstrained real-valued outputs:

$$\hat{y} = f(x; \theta), \hat{y} \in \mathbb{R}$$

How can we construct a binary classifier?

$$p(y = 1|x; \theta) = \text{Sigmoid}(f(x; \theta))$$

What is the training objective for classification problems?

$$\arg \min_{\theta} - \frac{1}{N} \sum_{n=1}^N [y_i \log p(y_i = 1|x_i; \theta) + (1 - y_i) \log(1 - p(y_i = 1|x_i; \theta))]$$

Quiz 1

Can you derive the training objective for classification problems with multiple classes?

Supervised Learning: Empirical Risk Minimisation

Hypothesis function class:

$$\mathcal{H} := \{f \mid f(x; \theta), \theta \in \mathbb{R}^d\}$$

Risk and expected loss:

$$R(f) = \mathbb{E}_{p(x,y)}[L(y, f(x))] \quad \begin{array}{l} \nearrow L(y, f(x)) = (y - f(x))^2 \\ \searrow \end{array}$$

Empirical risk:

$$L(y, f(x)) = y \log p(y = 1|x; \theta) + (1 - y) \log(1 - p(y = 1|x; \theta))$$

$$\hat{R}(f) := \frac{1}{N} \sum_{n=1}^N [L(y_n, f(x_n))]$$

Empirical risk minimisation:

$$\hat{f} = \arg \min_{f \in \mathcal{H}} \hat{R}(f) = \arg \min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^N L(y_n, f(x_n))$$

Generalisation

Empirical Risk:

$$\hat{R}(f) := \frac{1}{N} \sum_{n=1}^N [L(y_n, f(x_n))]$$

Population Risk:

$$R(f) = \mathbb{E}_{p(x,y)} [L(y, f(x))]$$

$$\hat{R}_{\text{test}}(f) = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} L(y, f(x))$$

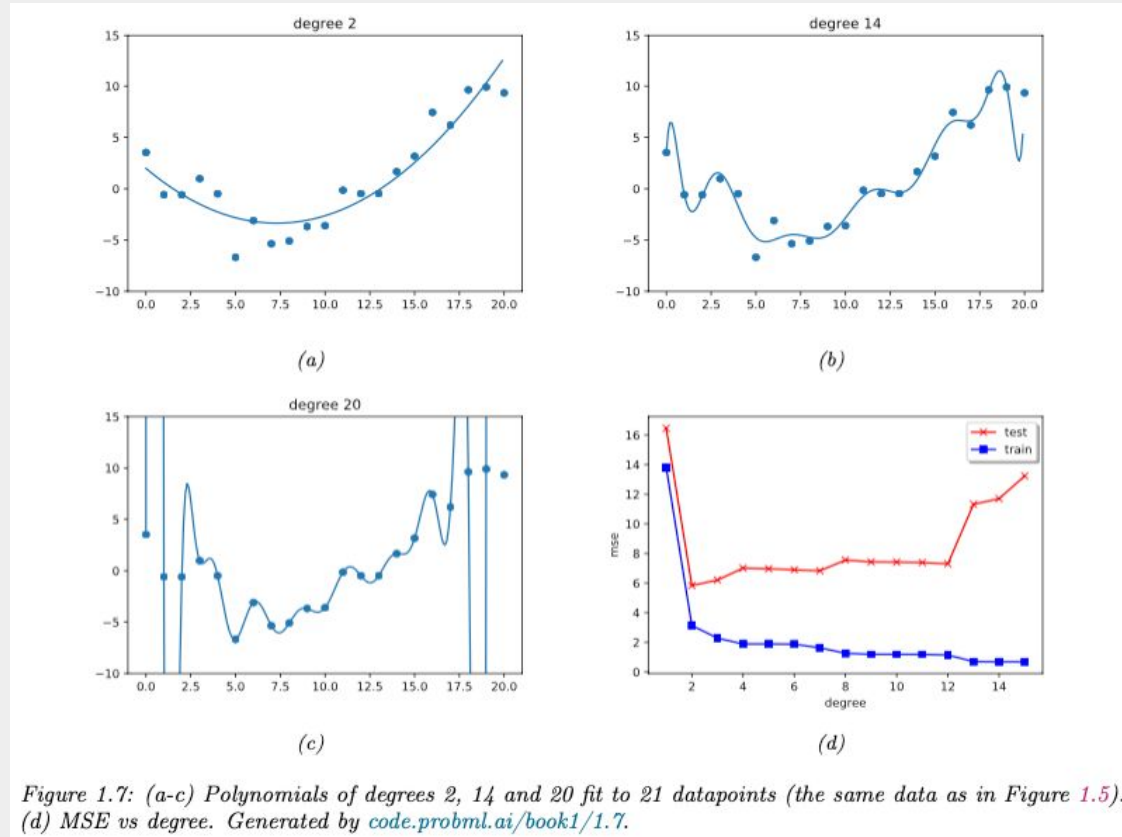
Generalisation gap:

$$R(f) - \hat{R}(f)$$

Generalisation: Bias & Variance

$$\begin{aligned}\mathbb{E}_{p(x,y)}[(y - \hat{f}(x))^2] &= \mathbb{E}[(f(x) + \epsilon - \hat{f}(x))^2] \\ &= \mathbb{E}[\epsilon^2] + \underbrace{\mathbb{E}[f(x) - \hat{f}(x)]^2}_{\text{Bias}^2} + \underbrace{\mathbb{V}[f(x) - \hat{f}(x)]}_{\text{Variance}}\end{aligned}$$

Generalisation: Overfitting & Underfitting



Regularisation

- Powerful models typically must be flexible and are therefore usually prone to overfitting
- **Regularization**: add a term $r(f)$, known as a **regularizer**, to the empirical risk that penalizes complex functions:

$$\hat{f} = \arg \min_{f \in \mathcal{H}} \hat{R}(f) + r(f) = \arg \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)) + r(f)$$

- Note that we would not need to regularize the true risk if we could calculate it: the job of the regularizer is to account for the overfitting bias induced by optimizing the empirical risk

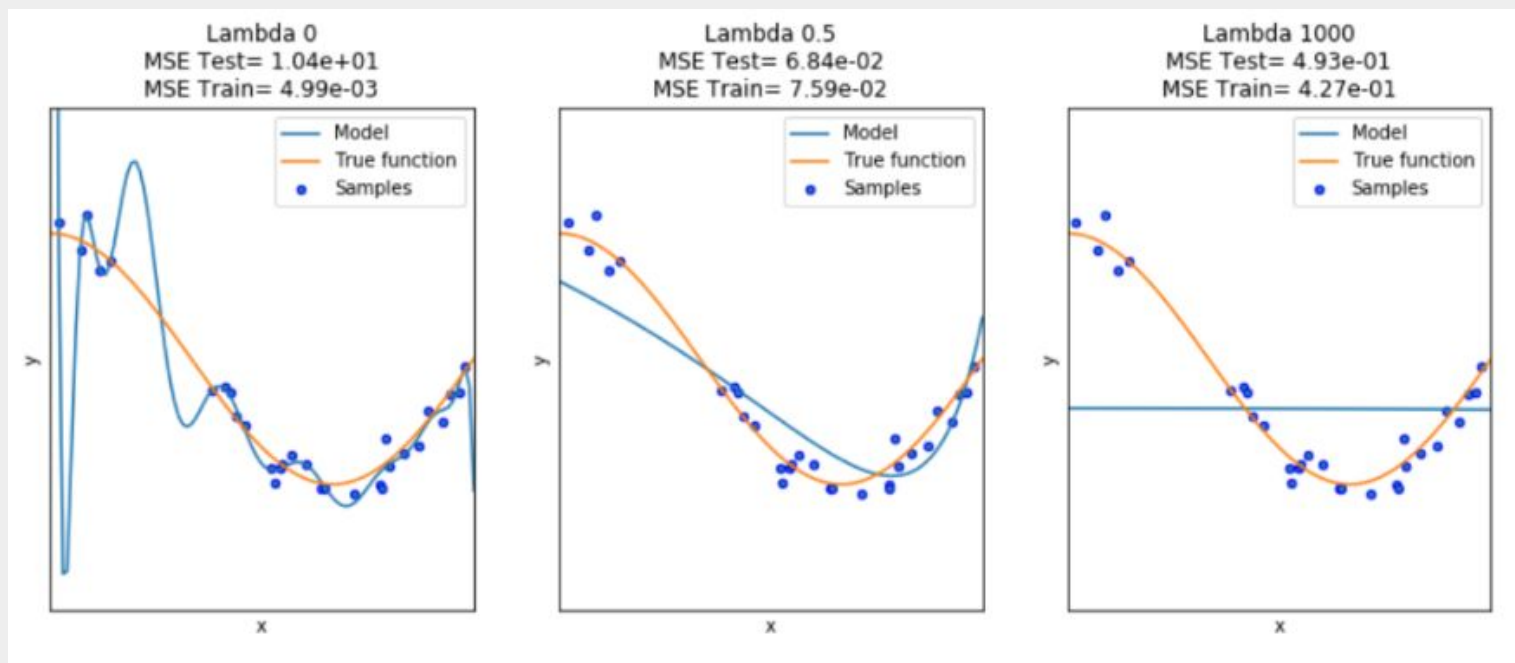
Regularisation

L_ρ norm regularisation:

$$\arg \min_{\theta} \hat{R}(f_{\theta}) + \lambda \|\theta\|_{\rho}^{\rho}$$

$$\text{where } \|\theta\|_{\rho} = \left(\sum_{j=1}^{\rho} |\theta_j|^{\rho} \right)^{1/\rho} \quad \rho \geq 1$$

Regularisation



Effect of L2 regularization

Learning Principles: A Probabilistic Perspective

Maximum Likelihood Estimation (MLE)

$$\arg \max_{\theta} \log p(\text{Data} \mid \theta)$$

Regularisation and Maximum A Posteriori (MAP)

$$\arg \max_{\theta} \log p(\text{Data} \mid \theta) + \log p(\theta)$$

Bayes rule: $p(\theta \mid \text{Data}) \propto p(\theta) \cdot p(\text{Data} \mid \theta)$

Quiz 2

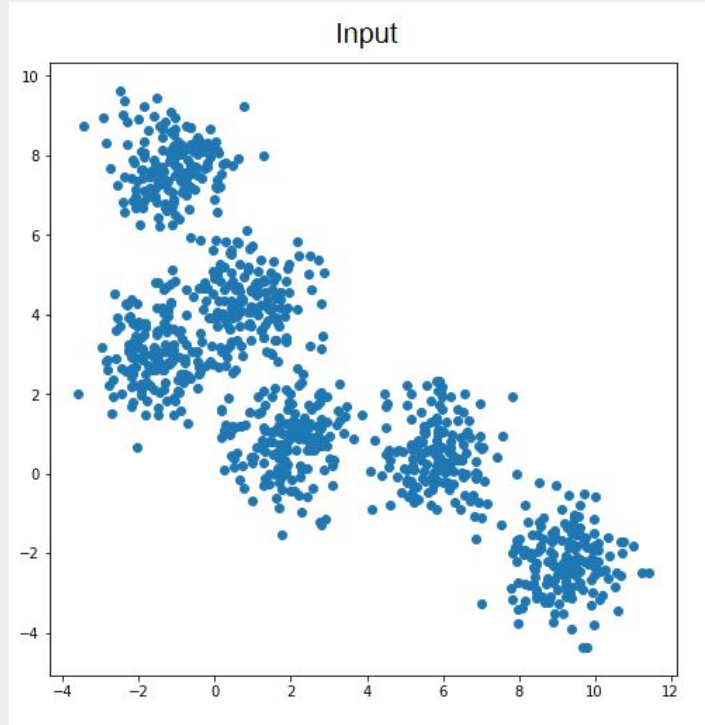
Maximum Likelihood Estimation (MLE)

Can you derive the training objective for regression task from the MLE principle?

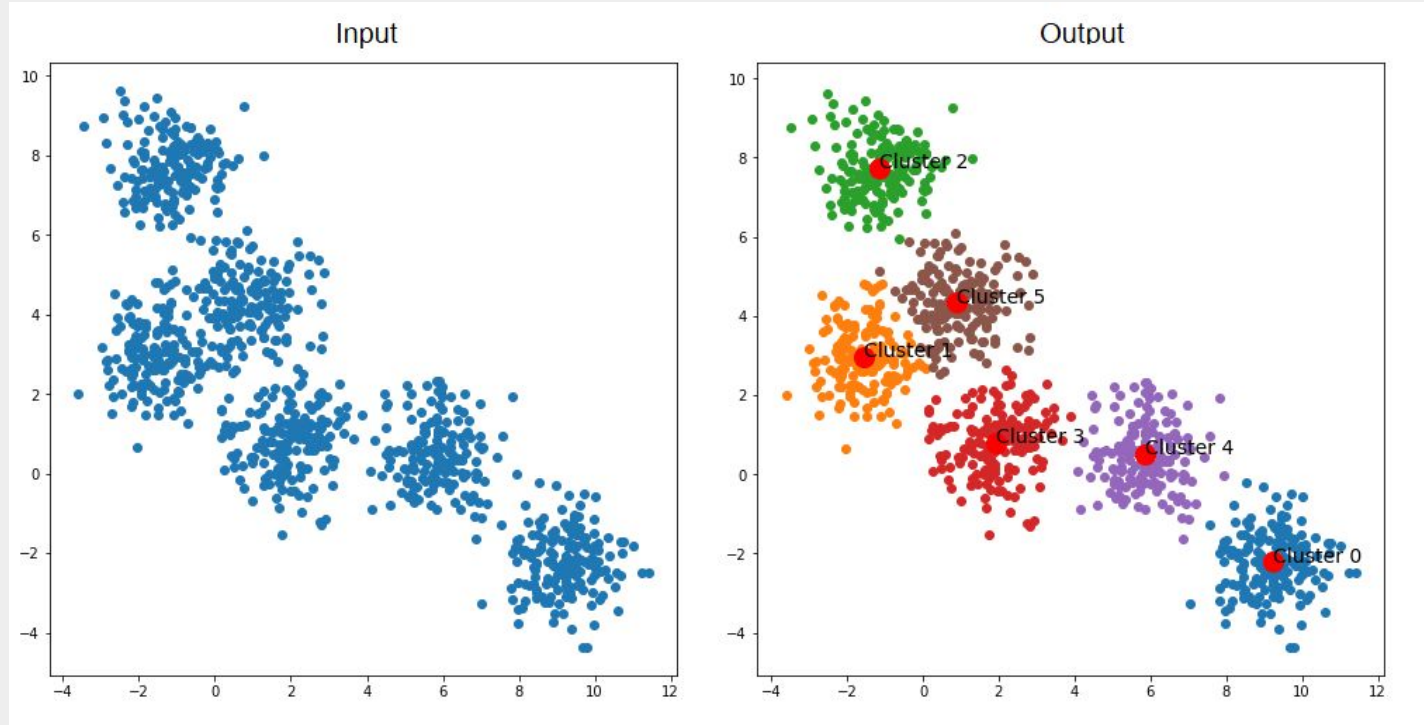
Regularisation and Maximum A Posteriori (MAP)

Can you connect L_2 regularisation with MAP learning principle?

Unsupervised Learning: Clustering

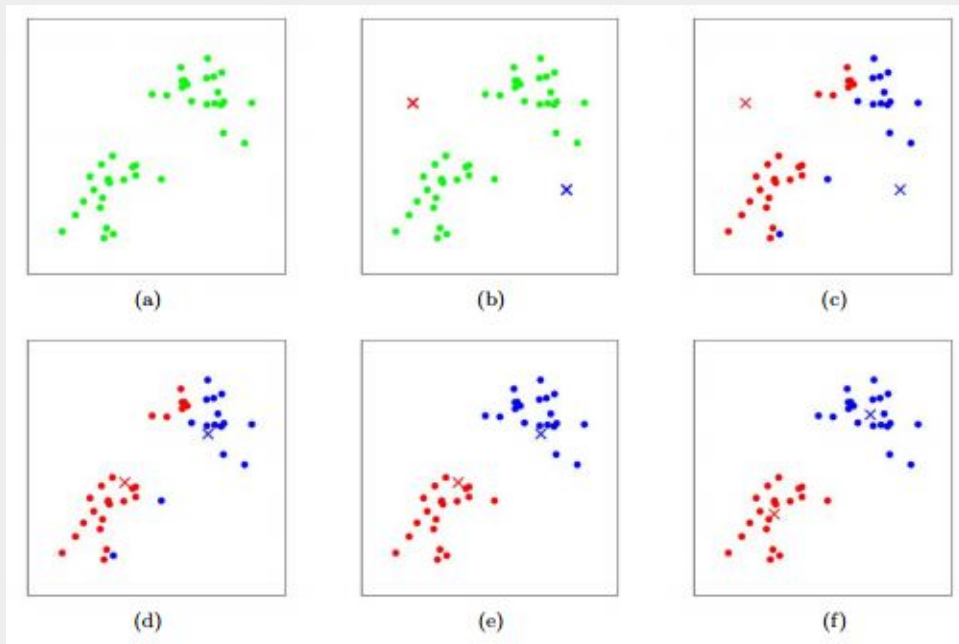


Unsupervised Learning: Clustering



Unsupervised Learning: Clustering

K-Means Algorithm:



1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

For every i , set

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2.$$

For each j , set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

Unsupervised Learning: Dimension Reduction

Ball on a frictionless spring recorded by 3 cameras:

- Imperfect measurements obfuscate true underlying dynamics.
- Our coordinates reflect method of data gathering rather than being meaningful themselves.

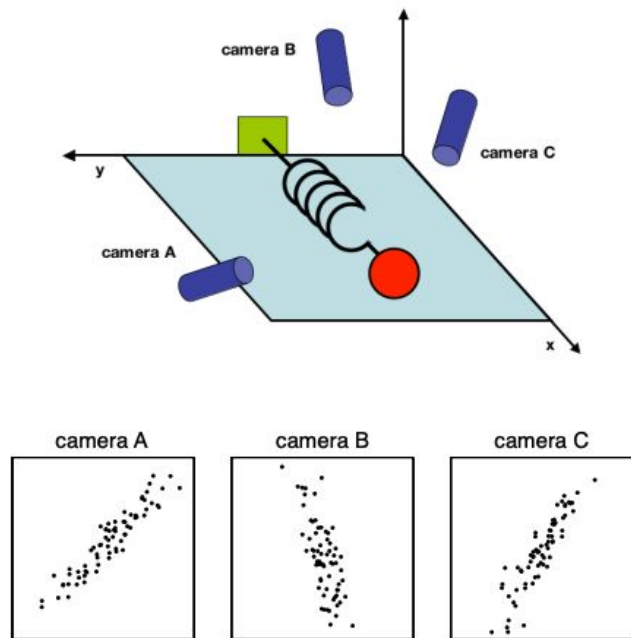
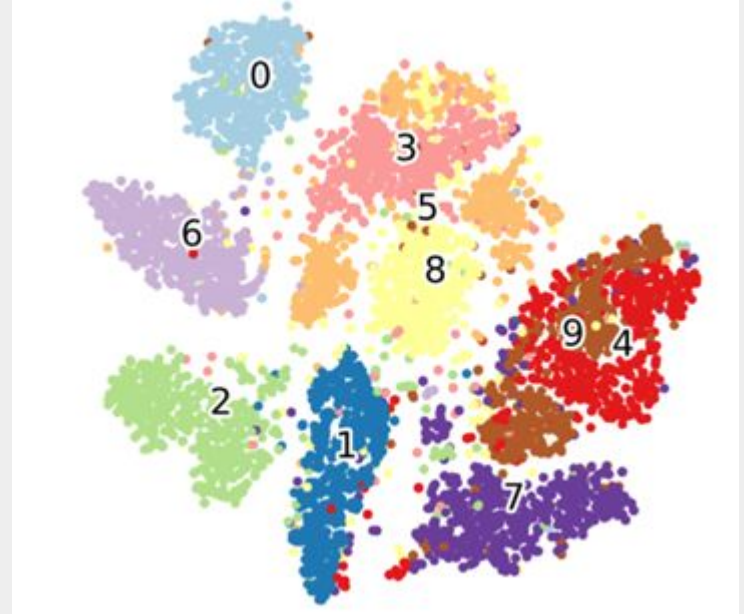
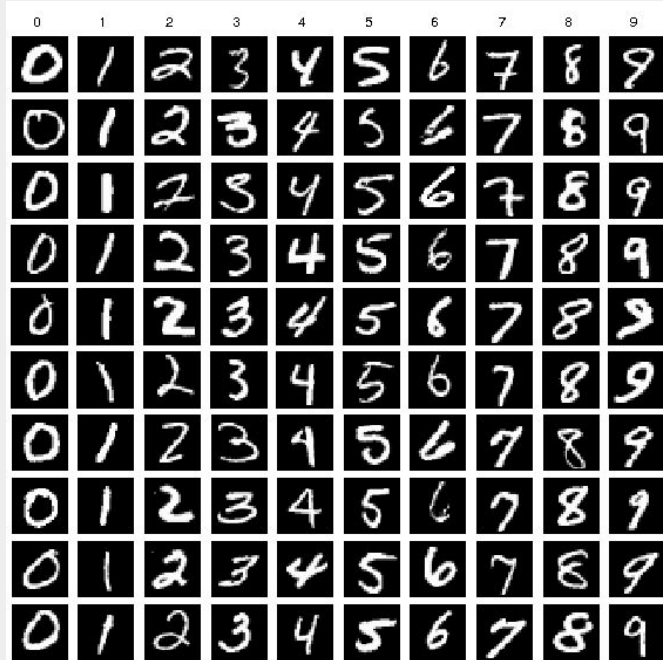


FIG. 1 A toy example. The position of a ball attached to an oscillating spring is recorded using three cameras A, B and C. The position of the ball tracked by each camera is depicted in each panel below.

Unsupervised Learning: Dimension Reduction

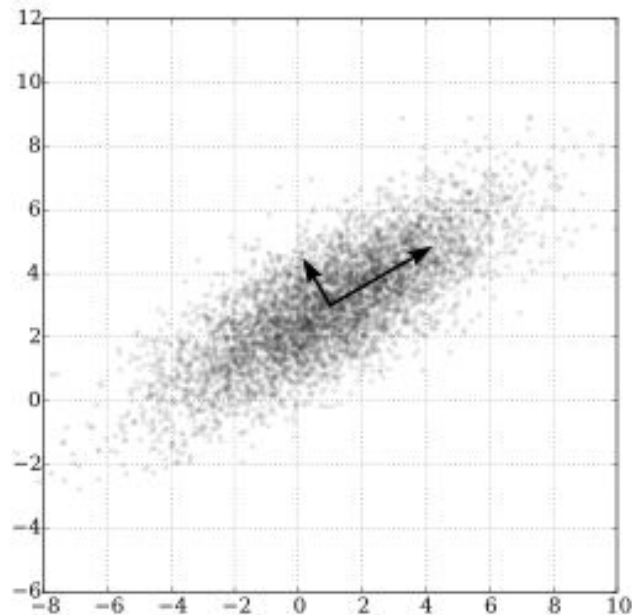


Unsupervised Learning: Dimension Reduction

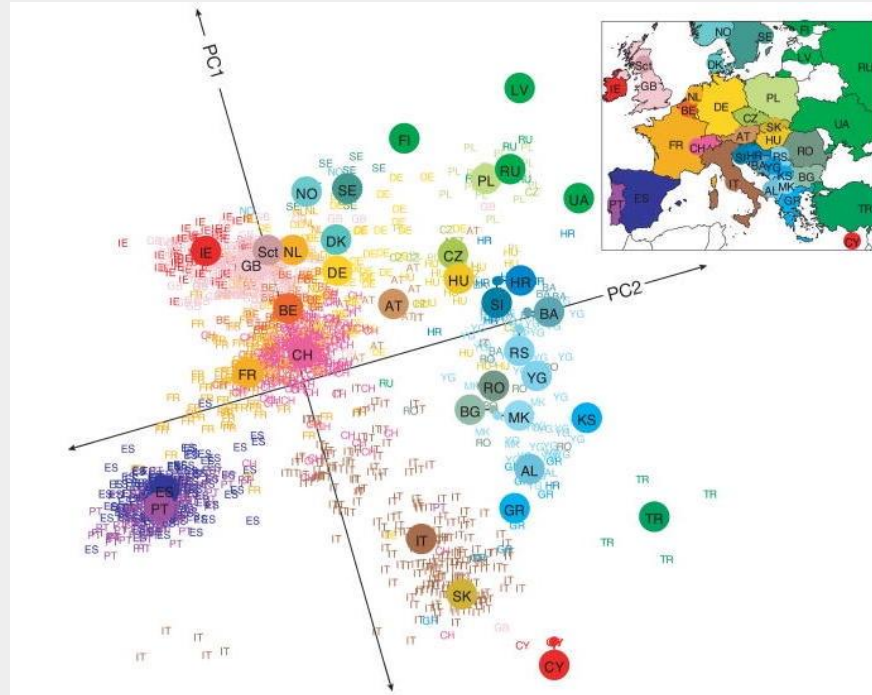
Principal Component Analysis (PCA):

The first principal component (PC) is the direction of greatest variance of data.

The j -th PC is the direction orthogonal to all previous PCs and is of greatest variance.



Unsupervised Learning: Dimension Reduction



Genes mirror geography within europe

Unsupervised Learning: Dimension Reduction

t-SNE(perplexity=10)



UMAP(n_neighbors=10)



TriMAP(n_inliers=8)



t-SNE(perplexity=20)



UMAP(n_neighbors=20)



TriMAP(n_inliers=10)



PaCMAP



t-SNE(perplexity=40)



UMAP(n_neighbors=40)



TriMAP(n_inliers=15)



¹ Van der Maaten, L.J.P.; Hinton, G.E. *Visualizing High-Dimensional Data*.

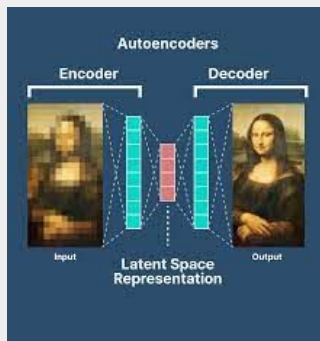
² McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*.

³ Amid, E., & Warmuth, M.K. (2019). *TriMap: Large-scale Dimensionality Reduction Using Triplets*. ArXiv, abs/1910.00204.

⁴ Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2021). *Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization*.

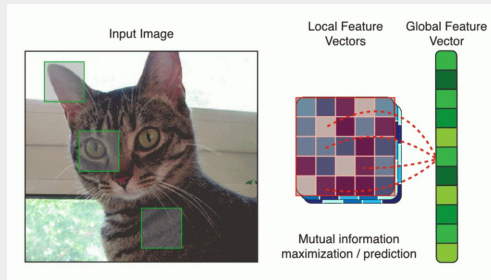
Unsupervised Learning: Representation Learning

A few examples of how one can learn representations from unlabelled data:



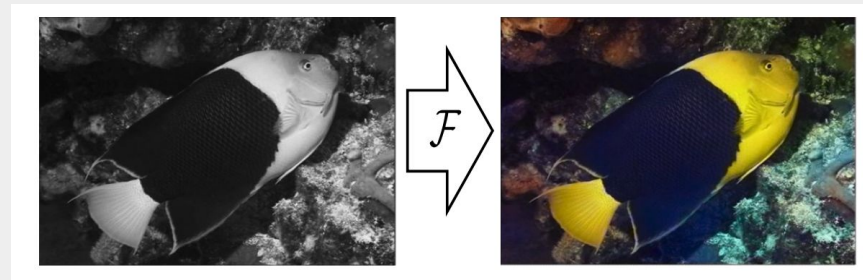
Autoencoders

Learning to reconstruct inputs through a low-dimensional latent representation.



Mutual Information Maximisation

Maximize the mutual information between inputs and representations.



Train model to predict colourisation

Make supervised learning problems from unsupervised learning problems. (self-supervised learning)

Unsupervised Learning: Density Estimation

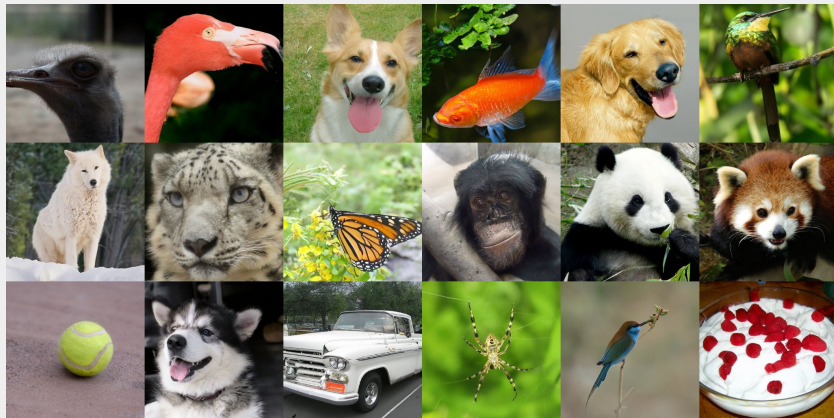
Given a set of samples $x_1, x_2, \dots, x_N \sim p(x)$

Approximate the true density $p(x)$ with density model $q_\theta(x)$

Unsupervised Learning: Density Estimation

With a learned density model $q_{\theta}(x)$, you can:

Sampling: $x \sim q_{\theta}(x)$



Evaluate density:

Given x^* compute $p(x^*)$

¹ Dhariwal, P., & Nichol, A. (2021). *Diffusion Models Beat GANs on Image Synthesis*.

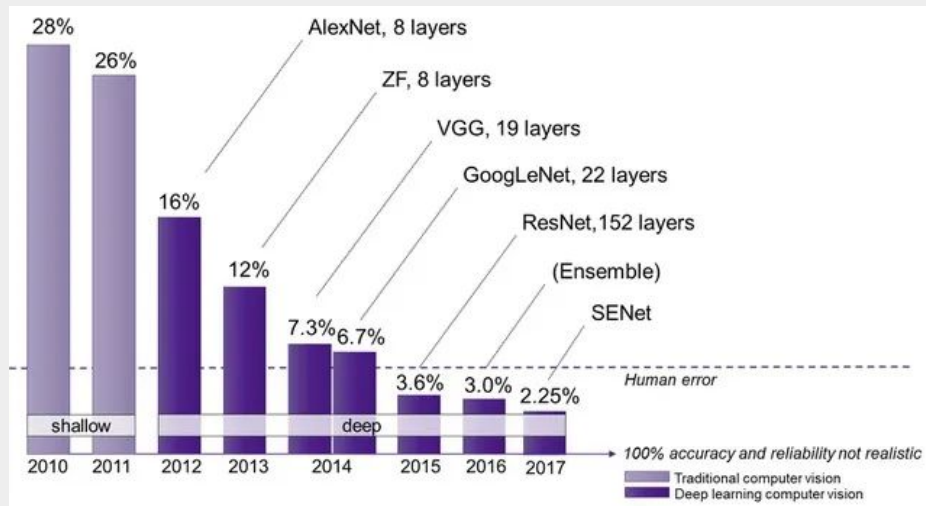
Other Types of Machine Learning

1. Semi-Supervised Learning.
 2. Reinforcement Learning.
 3. Active Learning.
 4. Online Learning.
 5. Meta Learning
-

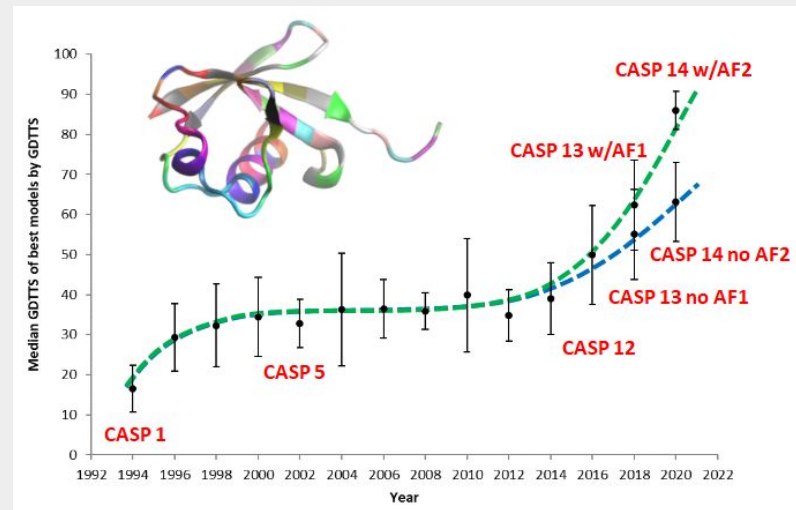
Data

Data plays a vital role in the success of machine learning.

Collecting data and creating benchmarks drives the progress of machine learning applications.



ImageNet Progress over the Years



The Critical Assessment of Structure Prediction (CASP)

Data: Tabular Datasets

Tabular Data

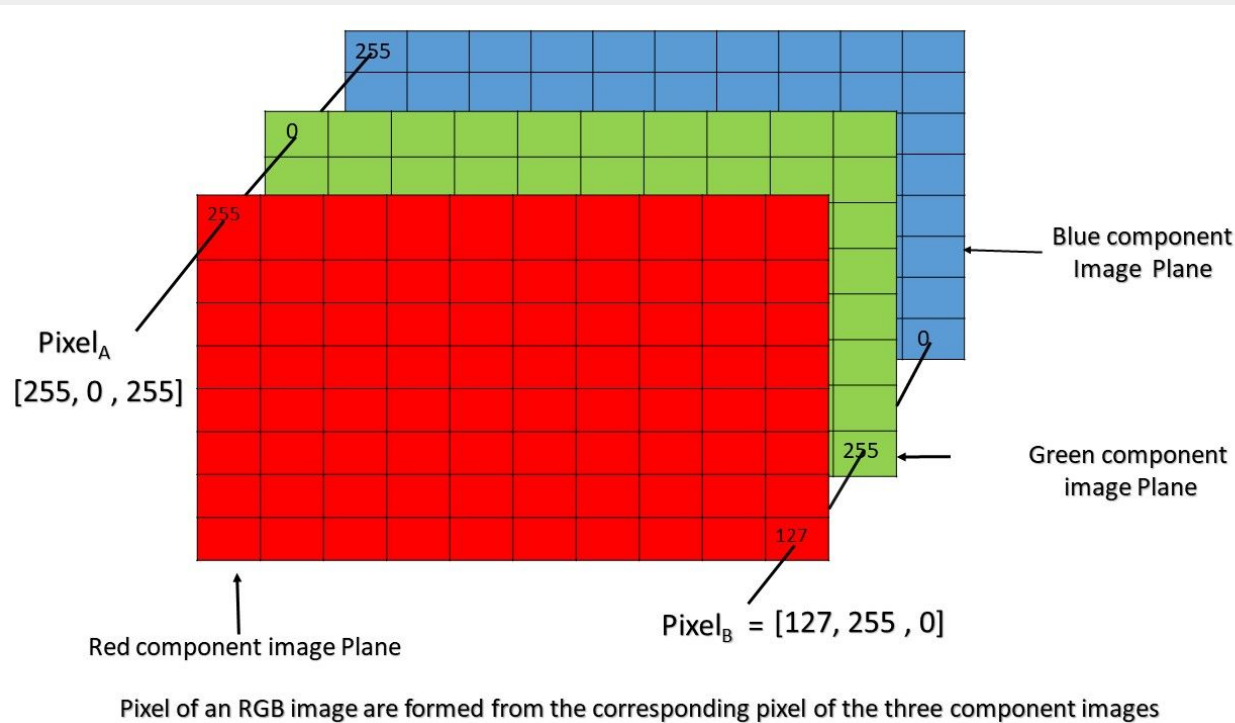
columns = attributes for those observations

Rows = observations

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

Matrix representation:
Shape $R \times C$

Data: Image Datasets

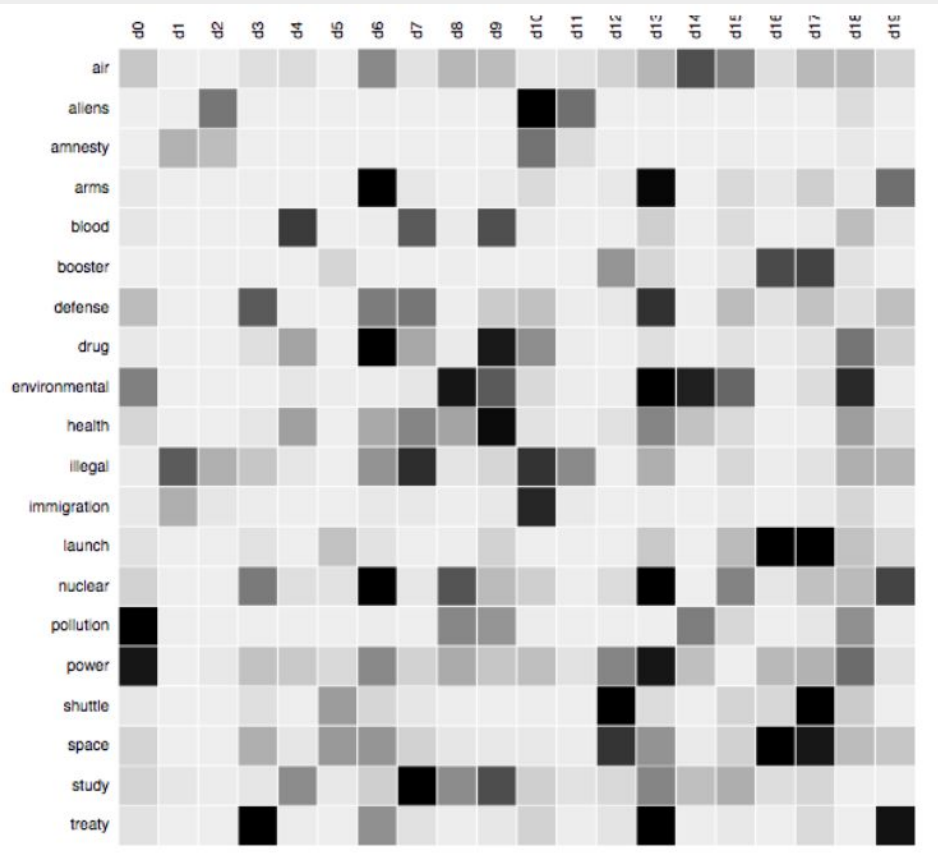


Tensor Representation:
Shape $B \times C \times H \times W$

Data: Text Datasets

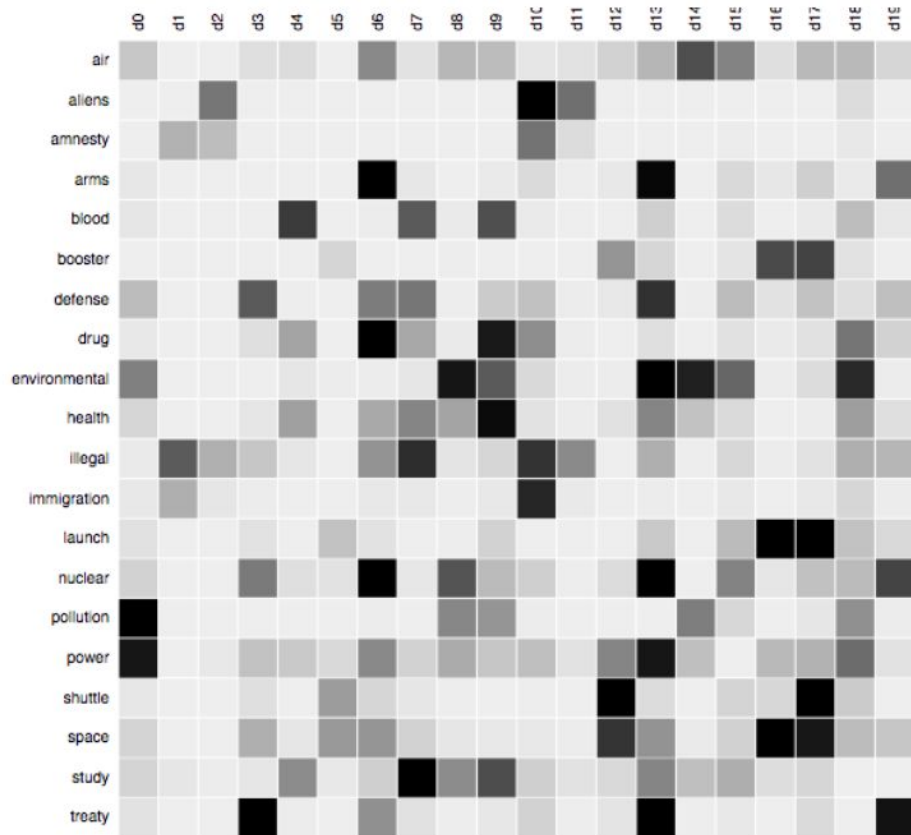
1. Documents have a variable length, and are thus not fixed-length feature vectors, as assumed by many kinds of models.
2. Words are categorical variables with many possible values (equal to the size of the vocabulary), so the corresponding one-hot encodings will be very high-dimensional.
3. We may encounter words at test time that have not been seen during training.

Bag of Words



$D \times N$ term frequency matrix,
where TF_{ij} is the frequency of
term i in document j .

Bag of Words



$D \times N$ term frequency matrix,
where TF_{ij} is the frequency of
term i in document j .

**TF-IDF (Frequency Matrix - Inverse
Document Frequency):**

$$TF\text{-}IDF_{ij} = \log(TF_{ij} + 1) \times IDF_i$$

$$IDF_i \triangleq \log \frac{N}{1 + DF_i}$$

where DF_i is the number of
documents with term i




Data: Processing Discrete Inputs

How to represent categorical features?













One-hot encoding!

Iris Type		Versicolor	Setosa	Virginica
Versicolor	→	1	0	0
Setosa		0	1	0
Virginica		0	0	1

Data: Handling Missing Data

	Feature 1	Feature 2	Feature 3
Sample 1			
Sample 2			
Sample 3			
Sample 4			

Data: Handling Missing Data

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
Sample 1				1	1	1
Sample 2				0	1	1
Sample 3				1	1	0
Sample 4				1	0	1

Binary Mask

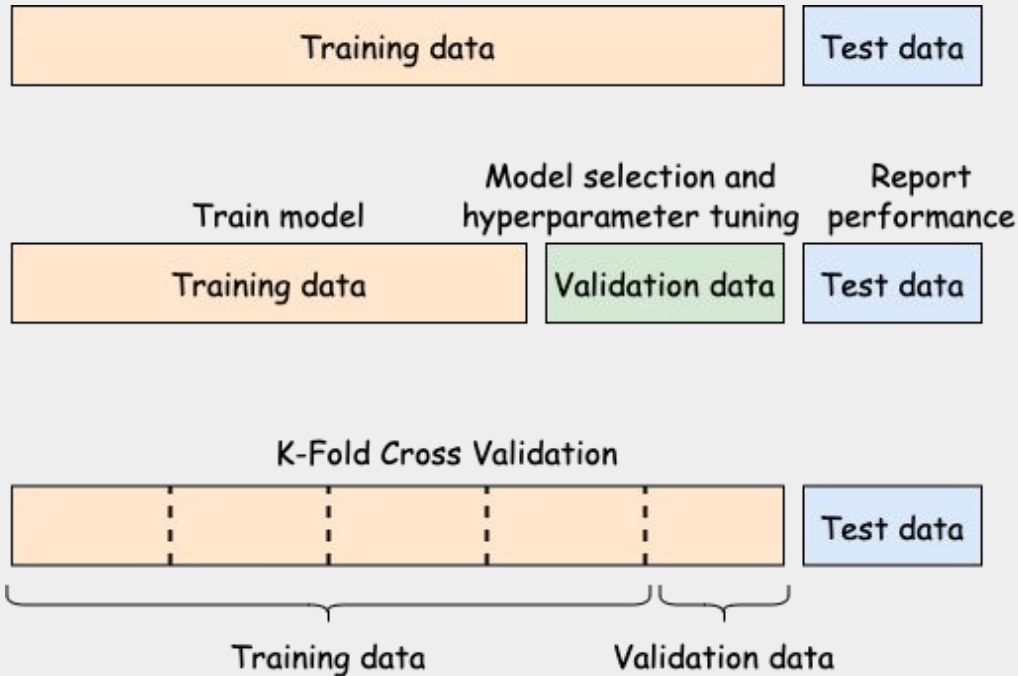
Data: Handling Missing Data

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6
Sample 1				1	1	1
Sample 2	?			0	1	1
Sample 3			?	1	1	0
Sample 4		?		1	0	1

Binary Mask

A common heuristic is called **mean value imputation**, in which missing values are replaced by their empirical mean

Training, Validation and Test Sets



Optimisation

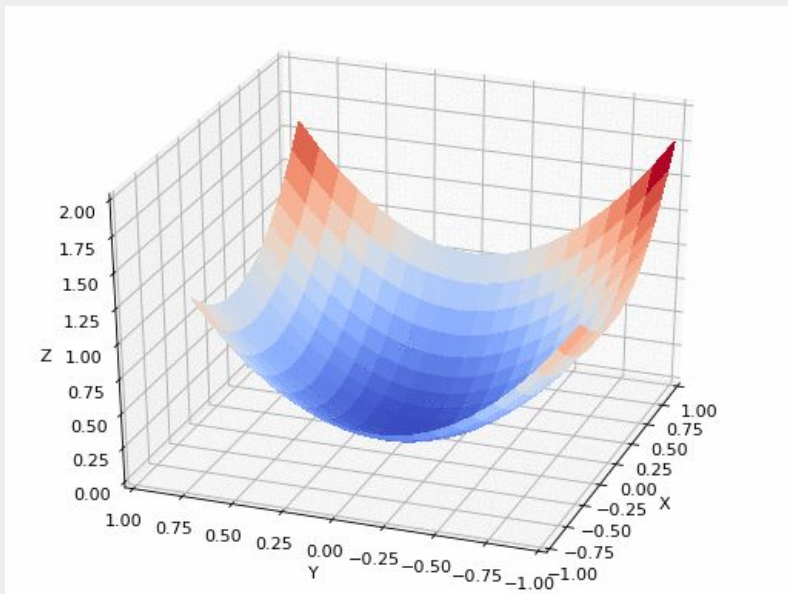
Optimisation plays a central role in training machine learning models and hyperparameter tuning.

There is a wide range of optimisation algorithms. And different optimisation methods are suitable for different problems and models.

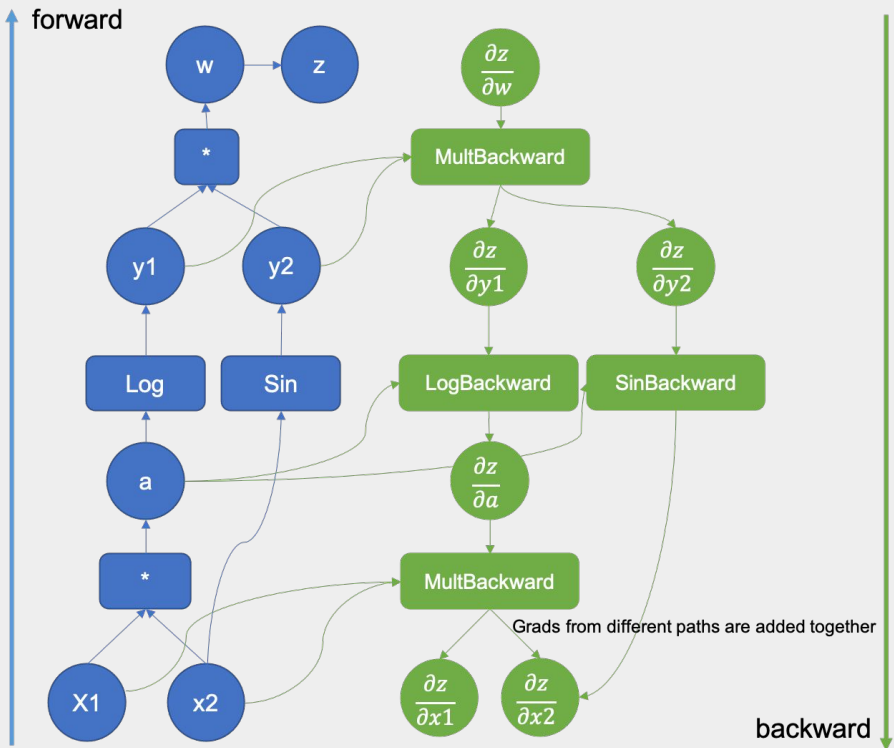
Optimisation: Gradient Descent

Solve $\arg \min_{\theta} \mathcal{L}(\theta)$

by iterative optimisation $\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} \mathcal{L}(\theta_t)$

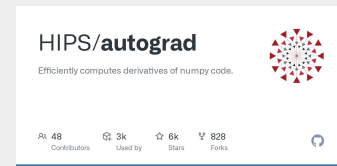


Optimisation: Automatic Differentiation



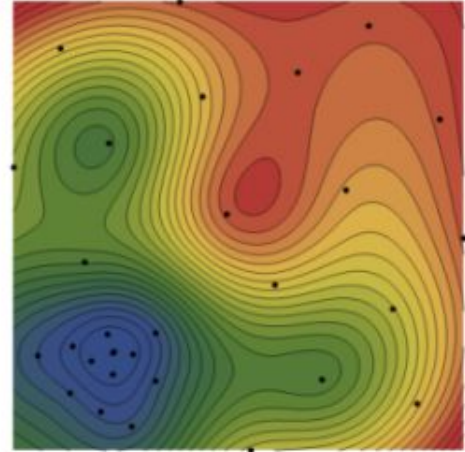
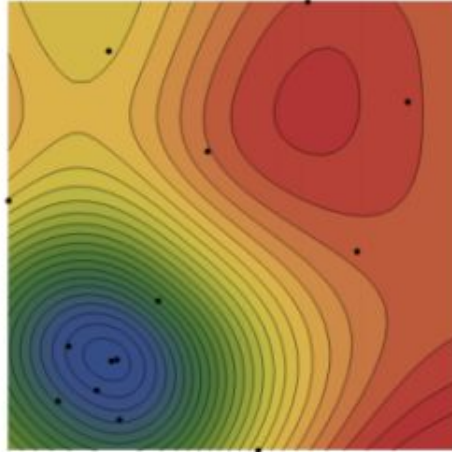
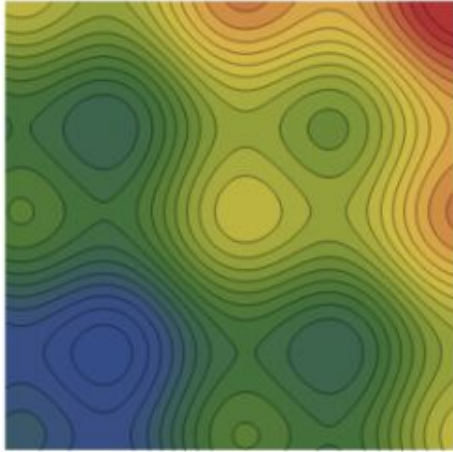
Thanks to automatic differentiation, we usually do not need to derive gradients for specific models anymore.

More details in the deep learning session!



Beyond First-Order Optimisation: Black Box Optimisation

Gradient-Free Optimisation.



¹ <https://github.com/paulknysh/blackbox>.

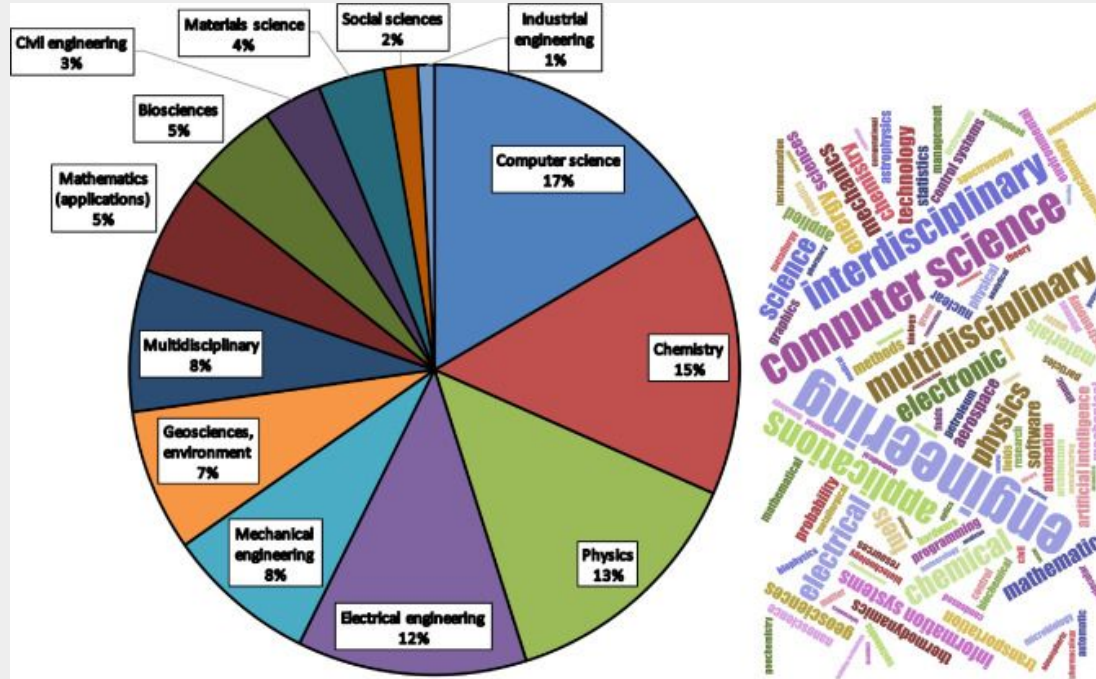
Beyond First-Order Optimisation:

Black Box Optimisation

Can be used for hyperparameter tuning!

Beyond First-Order Optimisation: Black Box Optimisation

Black-box optimisation are widely applied in all kinds of disciplines.

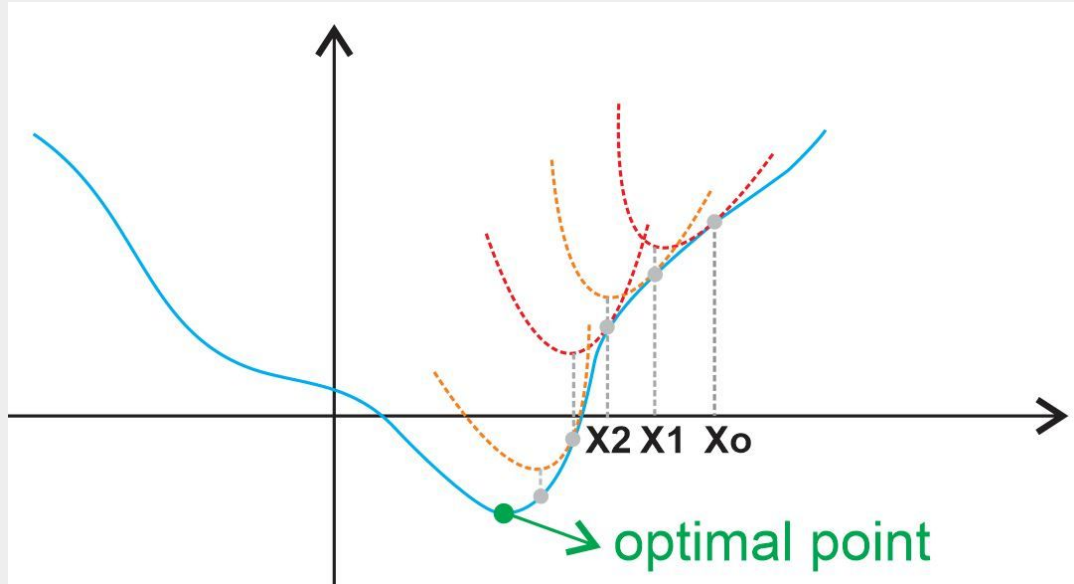


Distribution of the application fields that cite Digabel, S.L. (2011). Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm.

Beyond First-Order Optimisation:

Second-Order Optimisation Methods

Newton's Methods:



Second-order optimisation is normally computational expensive because we need to compute the inverse of the Hessian.

Approximate quasi-Newton methods that are memory or computational efficient are good alternatives.

¹ <https://ardianumam.wordpress.com/2017/09/27/newtons-method-optimization-derivation-and-how-it-works/>.

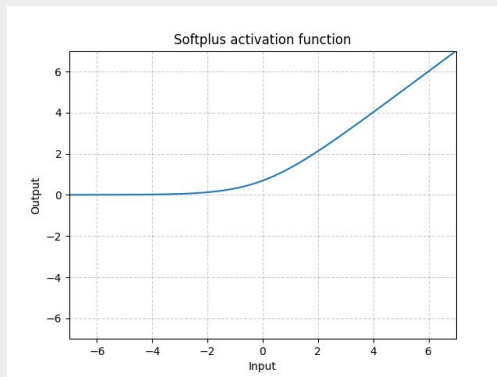
Constrained Optimisation to Unconstrained Optimisation

$$\min_{\theta} L(\theta) \text{ for } \theta > 0$$



$$\min_{\phi} L(\text{Softplus}(\phi))$$

$$\text{where } \text{Softplus}(\phi) = \frac{1}{\beta} \log(1 + \exp(\beta \cdot x))$$



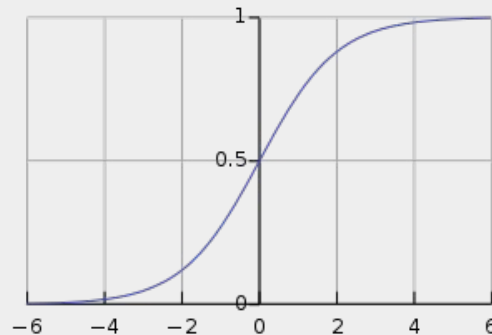
Softplus

$$\min_{\theta} L(\theta) \text{ for } 0 < \theta < C$$



$$\min_{\phi} L(C \cdot \text{Sigmoid}(\phi))$$

$$\text{where } \text{Sigmoid}(\phi) = \frac{1}{1 + \exp(-\phi)}$$



Sigmoid

No Free Lunch Theorem

No single best model that works optimally for all kinds of problems.

The set of assumptions that works well in one domain may work poorly in another.



*“Essentially,
all models are wrong,
but some are useful.”*

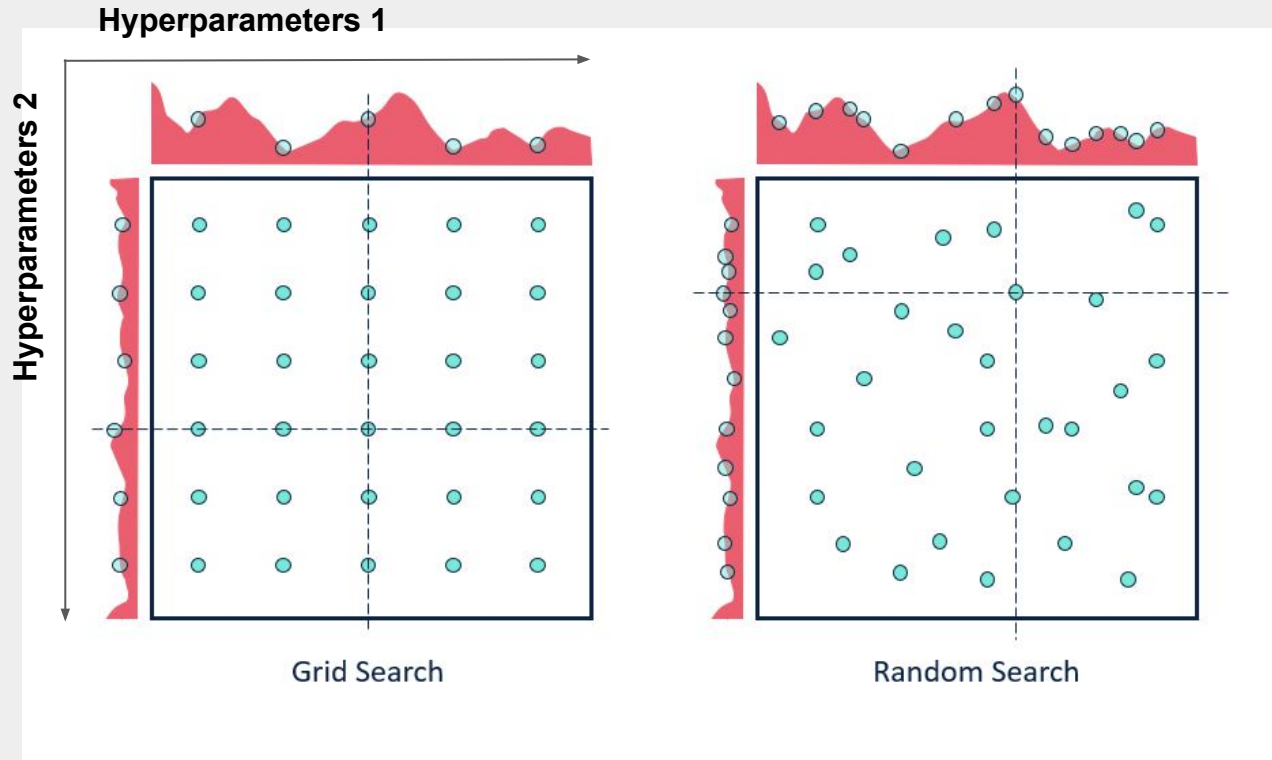
George Box
Statistician
1919-2013

Hyperparameter Tuning

Quiz 3

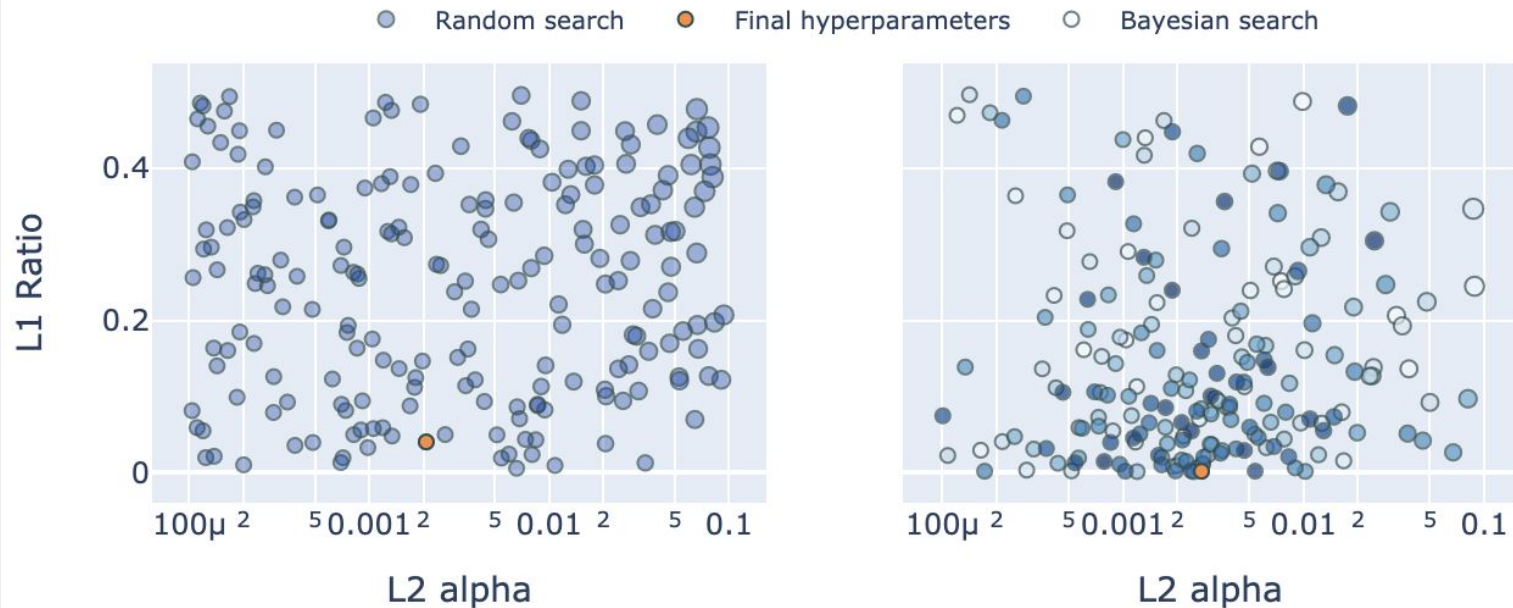
Why we can not optimise hyperparameters together with parameters?

Grid Search & Randomised Hyperparameter Search

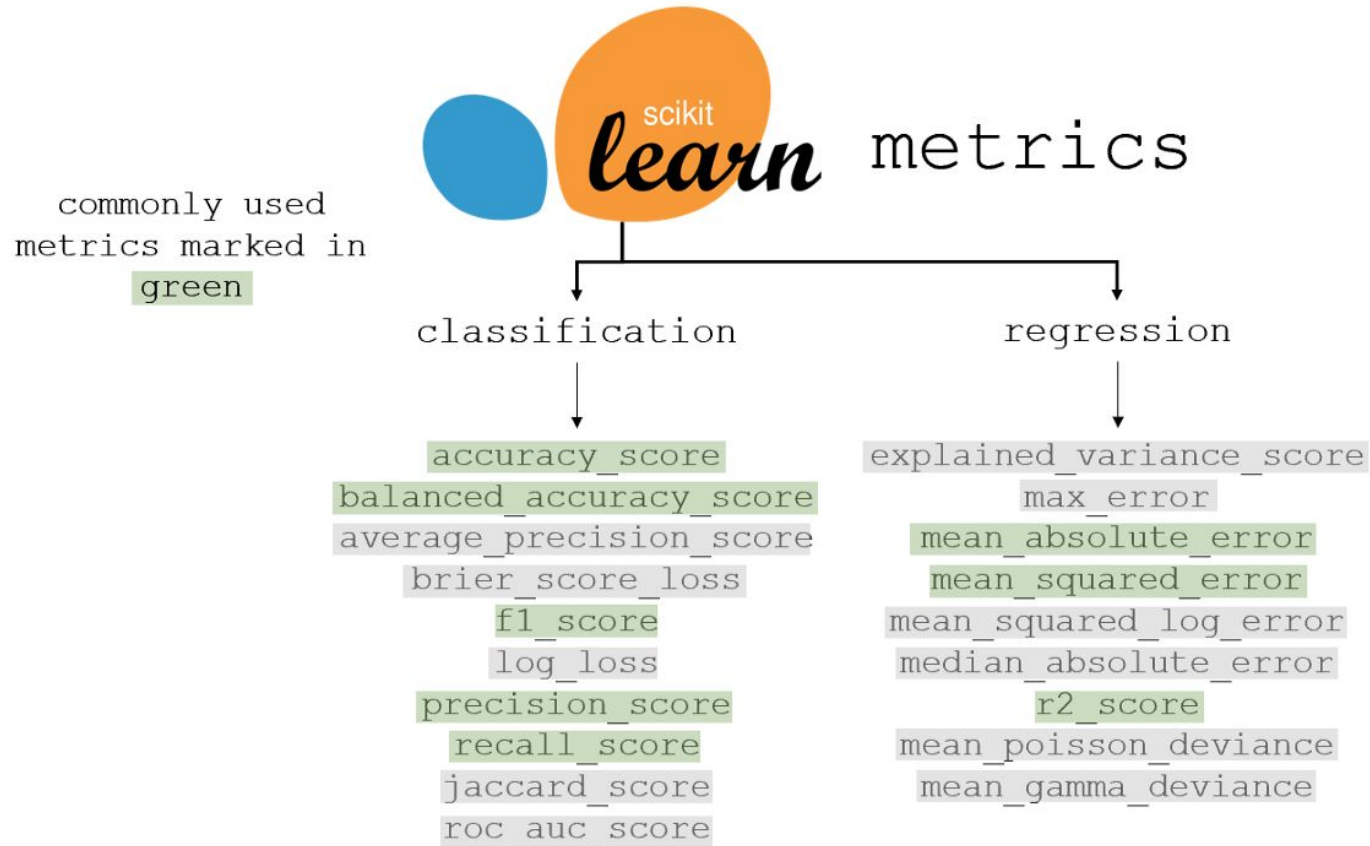


Bayesian Optimisation

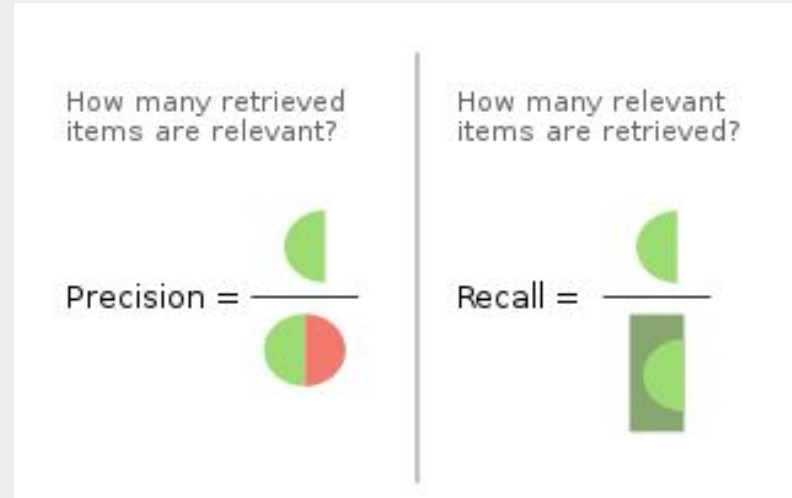
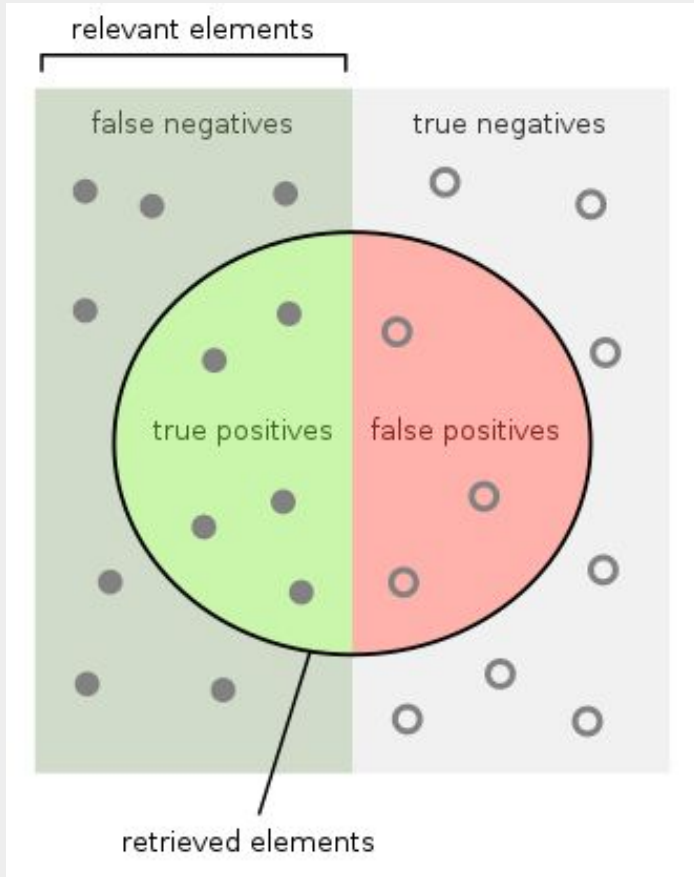
Random Search vs. Bayesian Search



Evaluation Metrics



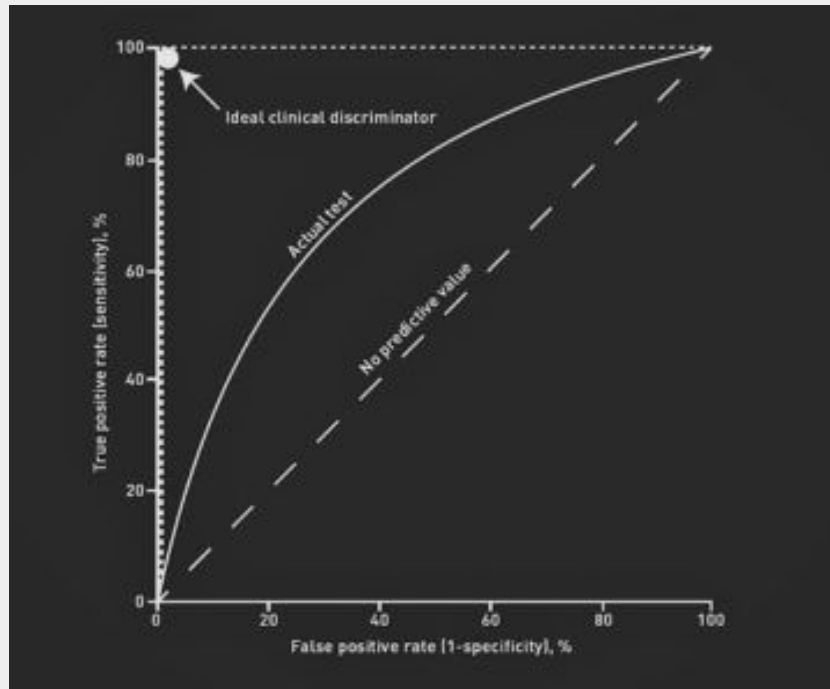
Evaluation Metrics: Precision, Recall & F1 Score



$$\text{F1 Score} = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}$$

Evaluation Metrics: ROC Curve

$$\text{TPR} = \frac{\text{True Positive}}{\text{All Positives}}$$



A random baseline classifier would give the diagonal.

Classifiers that give curves closer to the top-left corner indicate a better performance.

Source:

<https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/>

$$\text{FPR} = \frac{\text{False Positive}}{\text{All Negatives}}$$

Evaluation Metrics: Mean Square Error & R2 Score

Mean Squared Error:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

R2 Score:

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Thanks for Listening!

Q & A