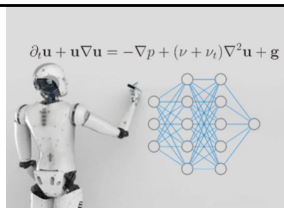


Hands on Machine Learning
for Fluid Dynamics
7 – 11 February 2022

von KARMAN INSTITUTE
FOR FLUID DYNAMICS

Lecture 1
What is Machine Learning ?
(and where does it meet physics?)

Miguel Alfonso Mendez
mendez@vki.ac.be

Table of Contents

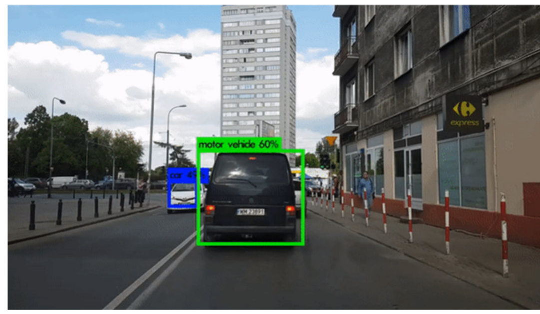
The Machine Learning Revolution: Data Analysis

Face Recognition on social media

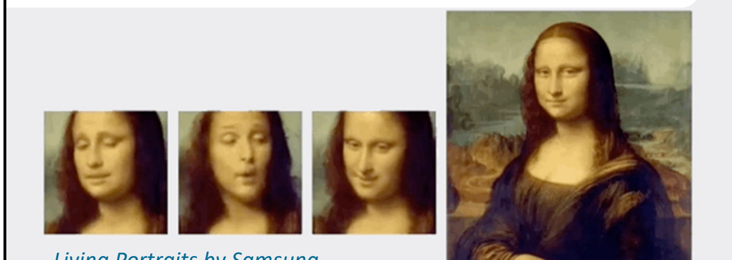
Real time audio translation

Japanese signs and menus instantly translated into English by Google Translate

Real Time Object Recognition



The Machine Learning Revolution: Data Generation



Living Portraits by Samsung

Deep Fakes using Generative Adversial Networks (GANS) and Autoencoders

An image generated by a StyleGAN:

This person does not exist

Look for Generative Adversial Networks (GANS) and fake images

https://www.whichfaceisreal.com/ or

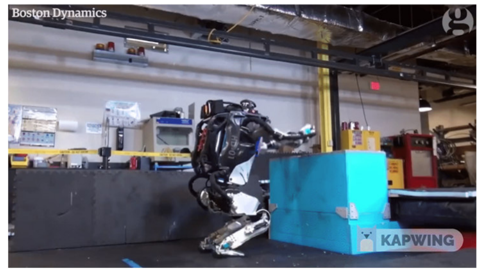
https://thispersondoesnotexist.com/

Slide 4

The Machine Learning Revolution: Taking Actions

DeepMind's AI parkour

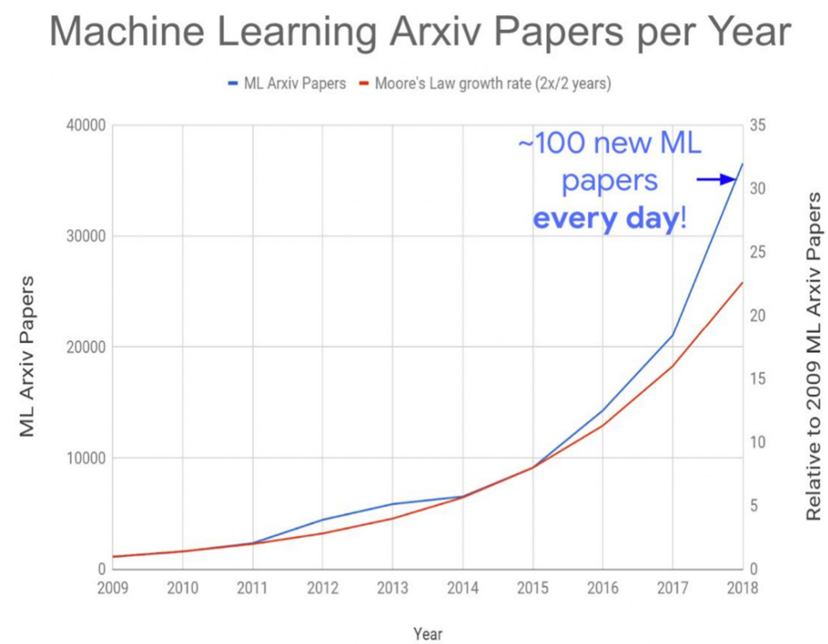
Atlas (Boston Dynamics) doing a backflip



Google's self driving car

Alpha Go Defeats Korean Go champion Lee Sedol

Slide 5



https://data-mining.philippe-fournier-viger.com/too-many-machine-learning-papers/ Slide 6

Machine Learning in Fluids

Invited

Perspective on machine learning for advancing fluid mechanics

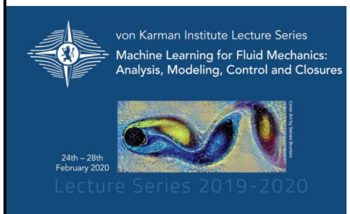
M. P. Brenner, J. D. Eldredge, and J. B. Freund
Phys. Rev. Fluids **4**, 100501 – Published 16 October 2019

Article | References | Citing Articles (26) | PDF | HTML | Export Citation

ABSTRACT

A perspective is presented on how machine learning (ML), with its burgeoning popularity and the increasing availability of portable implementations, might advance fluid mechanics. As with any numerical or experimental method, ML methods have strengths and limitations, which are acknowledged. Their potential impact is high so long as outcomes are held to the long-standing critical standards that should guide studies of flow physics.

Received 12 June 2019



Springer Link

Editorial | Published: 05 August 2020

Special issue on machine learning and data-driven methods in fluid dynamics

Steven L. Brunton, Maziar S. Hemati & Kunihiko Taira

Theoretical and Computational Fluid Dynamics **34**, 333–337(2020) | Cite this article

Machine learning (i.e., modern data-driven optimization and applied regression) is a rapidly growing field of research that is having a profound impact across many fields of science and engineering. In the past decade, machine learning has become a critical complement to existing experimental, computational, and theoretical aspects of fluid dynamics. In this short article, we are excited to introduce this special issue highlighting a number of promising avenues of ongoing research to integrate machine learning and data-driven techniques in the field of fluid dynamics. We will also attempt to provide a broader perspective, outlining recent successes, opportunities, and open challenges, while balancing optimism and skepticism.

ANNUAL REVIEWS

Annual Review of Fluid Mechanics

Machine Learning for Fluid Mechanics

Steven L. Brunton,[1] Bernd R. Noack,[2,3] and Petros Koumoutsakos[4]

[1]Department of Mechanical Engineering, University of Washington, Seattle, Washington 98195, USA

[2]LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur), CNRS UPR 3251, Université Paris-Saclay, F-91403 Orsay, France

[3]Institut für Strömungsmechanik und Technische Akustik, Technische Universität Berlin, D-10634 Berlin, Germany

[4]Computational Science and Engineering Laboratory, ETH Zurich, CH-8092 Zurich, Switzerland; email: petros@ethz.ch

Challenges and Opportunities for Machine Learning in Fluid Mechanics

M. A. Mendez, J. Dominique, M. Fiore, F. Pino, P. Sperotto, J. Van den Berghe

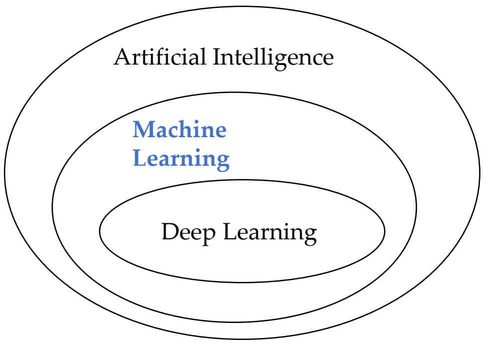
14 January 2022

Abstract

Big data and machine learning are driving comprehensive economic and social transformations and are rapidly re-shaping the toolbox and the methodologies of applied scientists. Machine learning tools are designed to learn functions from data with little to no need of prior knowledge. As continuous developments in experimental and numerical methods improve our ability to collect high-quality data, machine learning tools become increasingly viable and promising also in disciplines rooted in physical principles. These notes explore how machine learning can be integrated and combined with more classic methods in fluid dynamics. After a brief review of the machine learning landscape, we show how many problems in fluid mechanics can be framed as machine learning problems and we explore challenges and opportunities. We consider several relevant applications: aeroacoustic noise prediction, turbulence modelling, reduced-order modelling and forecasting, meshless integration of (partial) differential equations, super-resolution and flow control. While this list is by no means exhaustive, the presentation will provide enough concrete examples to offer perspectives on how machine learning might impact the way we do research and learn from data.

Table of Contents

What is Machine Learning ?

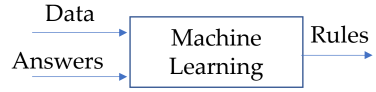


Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
-- Arthur Samuel, 1959

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E
-- Tom Mitchell, 1997

Old Programming Paradigm

New Programming Paradigm



The General Framework

Task: Learn a Target Function

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Experience: Training and Validation

$$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$$

$$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$$

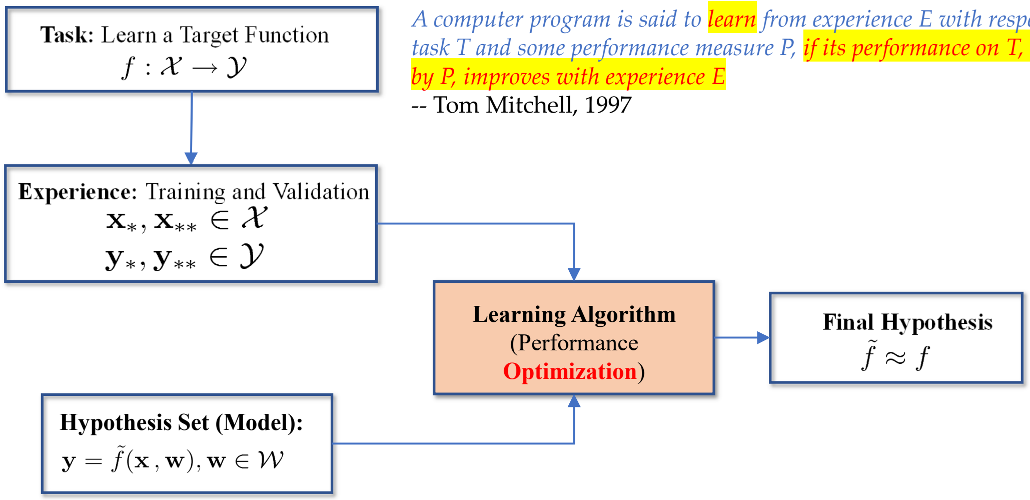
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E

-- Tom Mitchell, 1997

Note: * denotes training data, ** denotes validation data

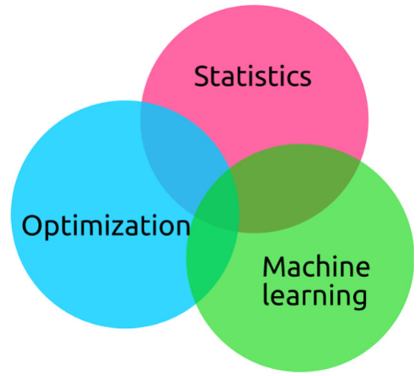
The General Framework

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E
-- Tom Mitchell, 1997



Note: * denotes training data, ** denotes validation data

The General Framework



Task: Learn a Target Function
$f : \mathcal{X} \rightarrow \mathcal{Y}$

Experience: Training and Validation
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$

Hypothesis Set (Model):
$\mathbf{y} = \tilde{f}(\mathbf{x}, \mathbf{w}), \mathbf{w} \in \mathcal{W}$

Learning Algorithm
(Performance **Optimization**)

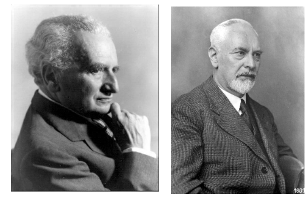
Final Hypothesis
$\tilde{f} \approx f$

Note: * denotes training data, ** denotes validation data

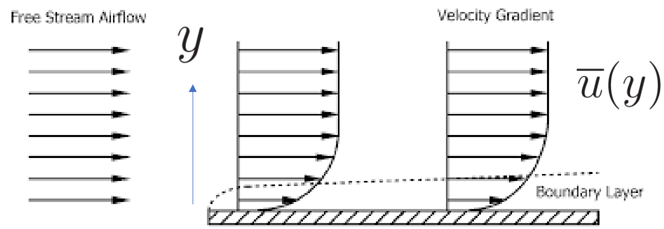
Machine learning…or Physics ?

Task:
look for a universal '*law of the wall*'

$$f : y \to \overline{u}(y)$$



Ludwig Prandtl and Theodore von Karman, ca 1930

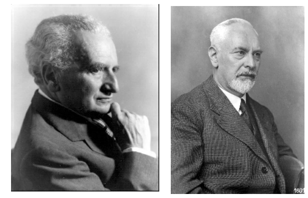


Machine learning…and Physics ?

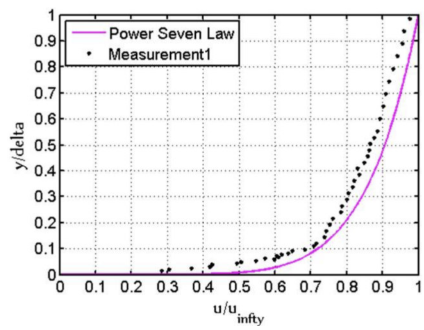
Task:
look for a universal '*law of the wall*'

$$f : y \to \overline{u}(y)$$

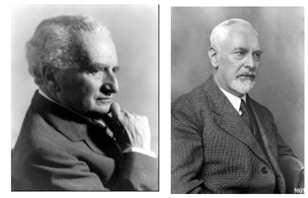
Experience:
Training and Validation data
$y_*, u_* \quad y_{**}, u_{**}$



Ludwig Prandtl and Theodore von Karman, ca 1930



Machine learning…and Physics



Ludwig Prandtl and
Theodore von Karman, ca 1930

Task:
look for a universal '*law of the wall*'

$$f : y \rightarrow \overline{u}(y)$$

Experience:
Training and Validation data
$y_*, u_* \quad y_{**}, u_{**}$

Hypothesis Set (Model):
$\mathbf{y} = \tilde{f}(\mathbf{x}, \mathbf{w}), \mathbf{w} \in \mathcal{W}$

$$\tau_t \approx \rho l^2 \left(\frac{d\overline{u}}{dy}\right) \approx \rho \kappa^2 y^2 \left(\frac{d\overline{u}}{dy}\right)$$

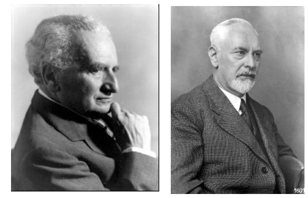
$$u^+(y^+) = \frac{1}{\kappa} \ln(y^+) + B$$

$$\tau_l = \rho\nu \frac{d\overline{u}}{dy}$$

$$u^+ = y^+$$

$$y^+ = \overline{y}\, u_\tau / \nu \qquad u^+ = \overline{u}/u_\tau \qquad u_\tau = \sqrt{\tau_w/\rho}$$

Machine learning…and Physics



Ludwig Prandtl and Theodore von Karman, ca 1930

Task:
look for a universal '*law of the wall*'

$$f : y \rightarrow \overline{u}(y)$$

Experience:
Training and Validation data

$$y_*, u_* \qquad y_{**}, u_{**}$$

$$\boxed{\overline{u} = f(y; \mathbf{w}) \qquad \mathbf{w} = [\kappa, B]^T}$$

Hypothesis Set (Model):
$\mathbf{y} = \tilde{f}(\mathbf{x}, \mathbf{w}), \mathbf{w} \in \mathcal{W}$

$$u^+(y^+) = \frac{1}{\kappa} \ln(y^+) + B \qquad \Big| \qquad u^+ = y^+$$

$$y^+ = \overline{y}\, u_\tau / \nu \qquad u^+ = \overline{u} / u_\tau \qquad u_\tau = \sqrt{\tau_w / \rho}$$

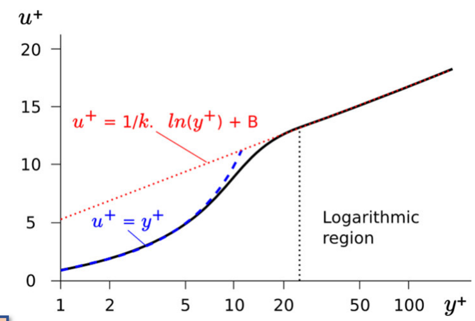
Machine learning…or Physics ?

Task:
look for a universal '*law of the wall*'

Experience:
Training and Validation data
$y_*, u_* \quad y_{**}, u_{**}$

Hypothesis Set (Model):
$\overline{u} = f(y; \mathbf{w}) \quad \mathbf{w} = [\kappa, B]^T$

Learning Algorithm
(Performance **Optimization**)

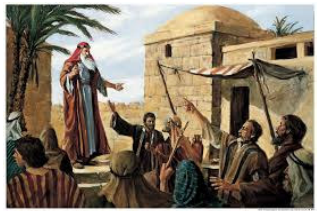


Final Hypothesis
$\kappa = 0.4 \quad B = 5$

Note: * denotes training data, ** denotes validation data

A note of Caution

Prophets:
Machine learning will soon do your job; you'll soon be jobless if you don't learn it quickly



Apostates:
Machine learning is just over-hyped curve fitting; the revolution will gradually die out and you better focus on something else

The **Lighthill report, 1973**

In no part of the field have the discoveries made so far produced the major impact that was then promised

The Lighthill debate on Artificial Intelligence: "The general purpose robot is a mirage"

Here's excerpt from an article in the New York Times 8 July 1958:
The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

A note of Caution

~~**Prophets**:
Machine learning will soon do your job; you'll soon be jobless if you don't learn it quickly~~

~~**Apostates**:
Machine learning is just over-hyped curve fitting; the revolution will gradually die out and you better focus on something else~~

Ordinary Least Squares

Linear Statistics and Regression

Fourier and Wavelet Analysis

Linear ROMs (POD, DMD, etc)

Somewhere in 2000-2010

Deep Neural Networks

Gaussian Processes

Autoencoders

Clustering Analysis

Bayesian Optimization

Somewhere in 2010-2020

Machine learning offers powerful tools and ignoring them means missing immense opportunities.

Machine learning is not the end goal of the fluid dynamists, and you must combine it with first principles.

Table of Contents

The General Framework

Machine Learning

Supervised Learning
(Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
$\mathbf{x}_{*}, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_{*}, \mathbf{y}_{**} \in \mathcal{Y}$

Regression **Classification**

Unsupervised Learning
(Descriptive)
$f : \mathcal{X} \rightarrow \mathcal{X}$
Experience:
$\mathbf{x}_{*}, \mathbf{x}_{**} \in \mathcal{X}$

Dimensionality Reduction **Clustering**

Semi-Supervised Learning

Reinforcement Learning

Note: * denotes training data, ** denotes validation data

The General Framework

Machine Learning

Supervised Learning
(Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$

- **Regression**
- **Classification**

Unsupervised Learning
(Descriptive)
$f : \mathcal{X} \rightarrow \mathcal{X}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

- **Dimensionality Reduction**
- **Clustering**

Semi-Supervised Learning
(Descriptive+ Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
'Some' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$
'a lot of' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

Reinforcement Learning
(Predictive, with critics !)
$\pi : \mathcal{S} \rightarrow \mathcal{A}$
Experience:
To be accumulated!

Note: * denotes training data, ** denotes validation data

The General Framework

Machine Learning

Supervised Learning
(Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$

- **Regression**
- **Classification**

Unsupervised Learning
(Descriptive)
$f : \mathcal{X} \rightarrow \mathcal{X}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

- **Dimensionality Reduction**
- **Clustering**

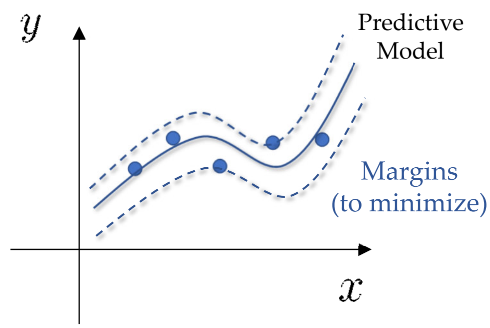
Semi-Supervised Learning
(Descriptive+ Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
'Some' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$
'a lot of' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

Reinforcement Learning
(Predictive, with critics !)
$\pi : \mathcal{S} \rightarrow \mathcal{A}$
Experience:
To be accumulated!

Note: * denotes training data, ** denotes validation data

Supervised Learning

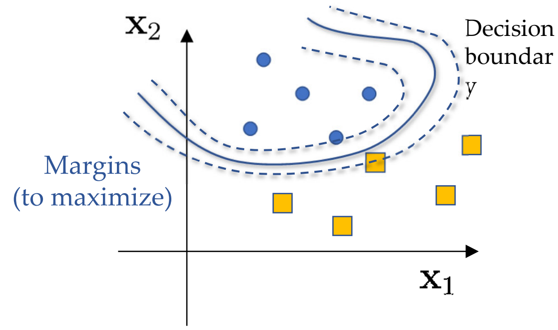
A regression problem:
Continuous output



Predict the cost of a car as a function of age

Predict the eddy viscosity in a point as a function of
Reynolds number and velocity gradient

A classification problem:
discrete (categorical output)

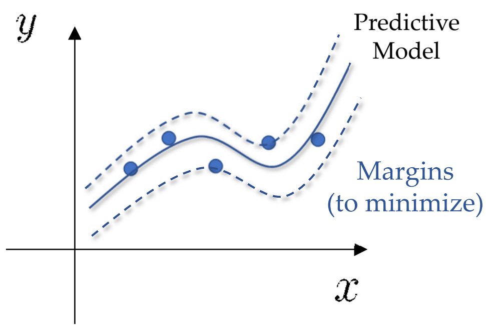


Handwritten character recognition

Classify flow regimes in a two-phase flow
as ‘bubbly’, ‘churn’, ‘annular’
(**with** a supervisor…!)

Supervised Learning

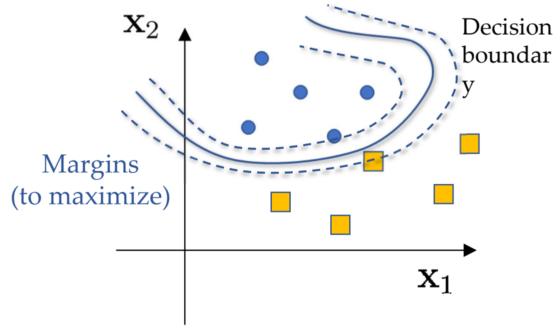
A regression problem:
Continuous output



Predict the cost of a car as a function of age

Predict the eddy viscosity in a point as a function of Reynolds number and velocity gradient

A classification problem:
discrete (categorical output)



Handwritten character recognition

Classify flow regimes in a two-phase flow as 'bubbly', 'churn', 'annular'
(**with** a supervisor…!)

The General Framework

Machine Learning

Supervised Learning
(Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$

- **Regression**
- **Classification**

Unsupervised Learning
(Descriptive)
$f : \mathcal{X} \rightarrow \mathcal{X}$
Experience:
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

- **Dimensionality Reduction**
- **Clustering**

Semi-Supervised Learning
(Descriptive+ Predictive)
$f : \mathcal{X} \rightarrow \mathcal{Y}$
Experience:
'Some' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$
'a lot of' $\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$

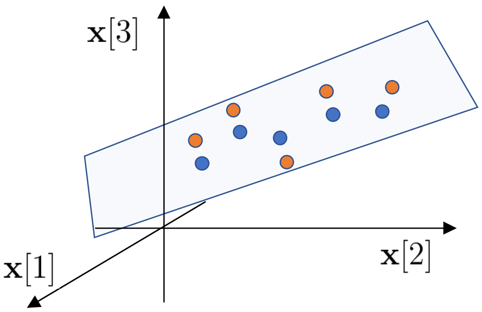
Reinforcement Learning
(Predictive, with critics !)
$\pi : \mathcal{S} \rightarrow \mathcal{A}$
Experience:
To be accumulated!

Note: * denotes training data, ** denotes validation data

Unsupervised Learning

A dimensionality reduction problem

(3 : 2)



Examples

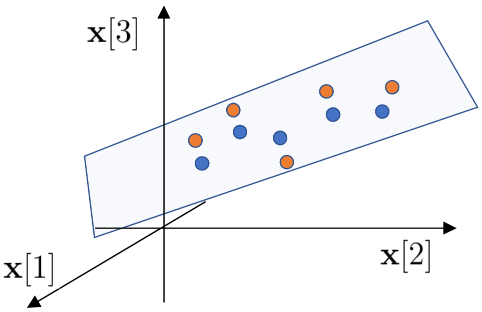
Image compression and noise removal

Predict the dynamics of Navier Stokes using ODEs (Reduced Order Modeling)

Slide 27

Unsupervised Learning

A dimensionality reduction problem
(3 : 2)

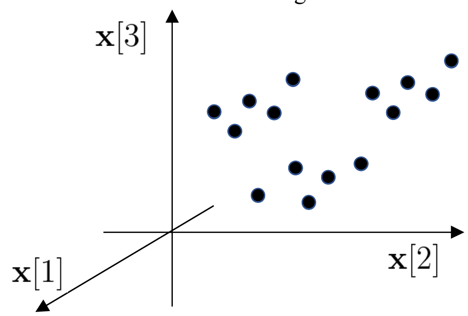


Examples

Image compression and noise removal

Predict the dynamics of Navier Stokes
using ODEs (Reduced Order Modeling)

A Clustering problem:
Careful: Clustering ≠ Classification



Examples

Group customers with similar behavior
Coarse graining an image for compression

Classify flow regimes in a two-phase flow
as ‘bubbly’, ‘churn’, ‘annular’
(**without** a supervisor…!)

Unsupervised Learning

A dimensionality reduction problem
(3 : 2)

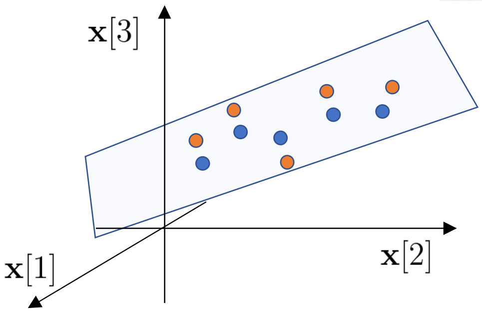
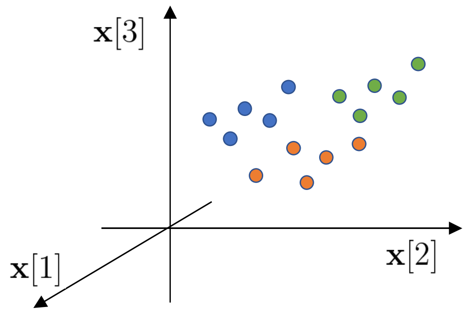


Image compression and noise removal

Predict the dynamics of Navier Stokes
using ODEs (Reduced Order Modeling)

A Clustering problem:
Careful: Clustering ≠ Classification

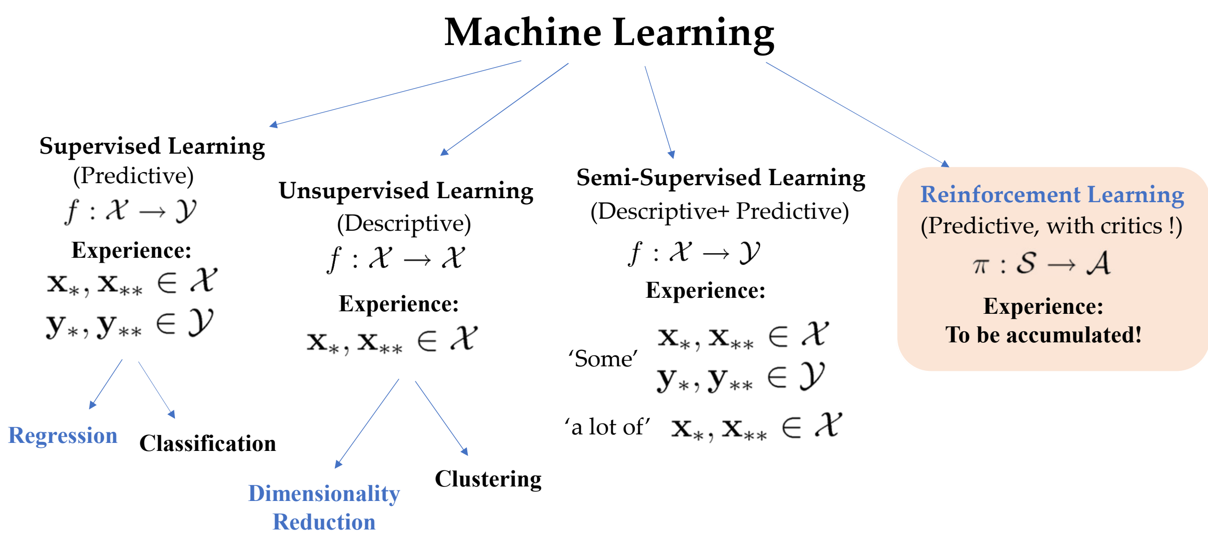


Group customers with similar behavior
Coarse graining an image for compression

Classify flow regimes in a two-phase flow
as ‘bubbly’, ‘churn’, ‘annular’
(without a supervisor…!)

Slide 29

The General Framework



Note: * denotes training data, ** denotes validation data

Reinforcement Learning

Environment and Agent are
Markov Decision Process (MDP)

Episodes $\tau = \{(s_0, a_0, r_0), (s_1, a_1, r_1), \dots\}$

Global Rewards $R_\tau = \sum_\tau \gamma^k r_t$

How good it is to be in s_t
Value Function :
$V^\pi(s) = \mathbb{E}[R_t | s, \pi]$

How good it is to do a_t in s_t
State-Value Function :
$Q^\pi(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t, \pi]$

Agent
$a_t = \pi(s_t, \mathbf{w}), \mathbf{w} \in \mathcal{W}$

s_t r_t a_t

Environment
$(s_{t+1}, r_{t+1}) = P(s_t, r_t)$

Objective:

Policy-based: try to learn $a_t = \pi(s_t, \mathbf{w})$

Off policy: try to learn $Q^\pi(s_{,t}\,, a_t)$

Then, take actions accordingly

Example:
a chessboard
game

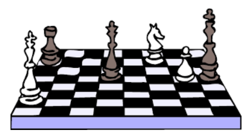


Table of Contents

The most elementary ML task… revised

We consider the most elementary machine learning task: a linear regression, i.e. the model

$$y(x) = w_1 x + w_0 + y_\epsilon \quad \text{with} \quad y_\epsilon \sim \mathcal{N}(y; 0, \sigma_y^2)$$

More generally, a single (real valued) variable Gaussian will be $\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{x-\mu}{2\sigma^2}\right)}$

In vector notation, the prediction at a given location is

$$y(x) = \mathbf{w}^T X + y_\varepsilon = \underbrace{f(x; \mathbf{w})}_{\text{Function to learn}} + y_\epsilon \quad \text{with} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \; X = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

In matrix notation, the prediction on a set $\mathbf{x} = [x_0, x_1, \dots x_{n_p}]^T$ $\mathbf{y} = [y_0, y_1, \dots y_{n_p}]^T$

$$\mathbf{y} = \mathbf{X}\mathbf{w} \quad \text{With the design matrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_0 \\ | & | \\ 1 & x_{n_p-1} \end{bmatrix} \in \mathbb{R}^{n_p \times 2}$$

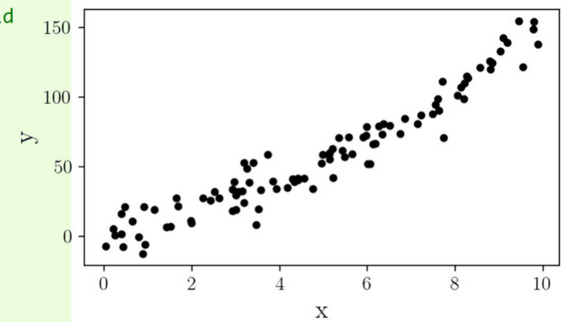
A toy example

To practice in today's exercise, here is how you can create a plot a dataset

```
# Generate data
n_p=100
np.random.seed(10)
x_s=np.random.uniform(0,10,n_p)
y_s=2*x_s+2+np.random.normal(loc=0,scale=10,size=len(x_s))+x_s**2
```

Note the quadratic term and the large noise level.

```
# Here's the fit function from numpy
w_s=np.polyfit(x_s,y_s,1)
# We now want to test our model on a new regular grid
x_t=np.linspace(0,10,200)
# The prediction would thus be
y_t=np.polyval(w_s,x_t)
# Show the result of the fit
fig, ax = plt.subplots(figsize=(5, 3))
plt.scatter(x_s,y_s,c='black',
            marker='o',edgecolor='black',s=16)
plt.plot(x_t,y_t,'r--',linewidth=2)
ax.set_xlabel('x',fontsize=16)
ax.set_ylabel('y',fontsize=16)
Name='Exercise_1_data_fit.png'
plt.tight_layout()
plt.savefig(Name, dpi=200)
```

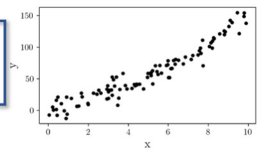


Slide 34

Table of Contents

The training problem

Task: Learn a Target Function
$f : \mathcal{X} \rightarrow \mathcal{Y}$



Experience: Training and Validation
$\mathbf{x}_*, \mathbf{x}_{**} \in \mathcal{X}$
$\mathbf{y}_*, \mathbf{y}_{**} \in \mathcal{Y}$

Learning Algorithm
(Performance **Optimization**)

Hypothesis Set (Model):
$y(x) = \mathbf{w}^T X + y_\varepsilon$

Final Hypothesis
$y(x) = w_1 x + w_0 + y_\epsilon$

Three perspectives

(1) 'Common sense' and calculus

(2) Probabilistic, in a 'frequentist' sense

(3) Probabilistic, in a `Bayesian' sense

We see it on Wednesday ☺

View 1: 'Common sense' or 'noise minimization'

If we are to make predictions, we must hope that the stochastic part is 'small'.

If we believe that our univariate Gaussian model for the random noise is valid, we should find the weights according to which the 'unpredictable part' is the smallest possible.

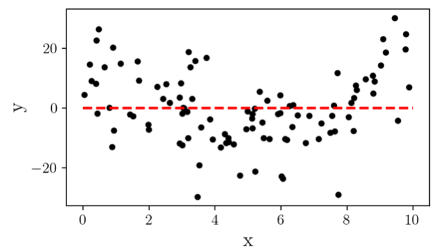
If we define a 'cost function' of the form

$$J(\mathbf{w}) = \frac{1}{n_p}\sum_{j=0}^{n_p-1}\Big(f(x_i;\mathbf{w}) - y_j\Big)^2 = \frac{1}{n_p}\sum_{j=0}^{n_p-1}\big(r_j\big)^2 = \frac{1}{n_p}||\mathbf{r}||_2^2$$

We should have: $\underset{\mathbf{w}}{\operatorname{argmin}}\, J(\mathbf{w})$

Note that this cost function is essentially an estimate of

$$\boxed{\sigma_y \approx \sqrt{J(\mathbf{w})}}$$



View 2: the frequentist view and MLE

Assume that each of those points are independent and identically distributed (i.i.d).

What is the probability that you observe exactly **that** sequence of numbers?

The probability of observing one point is given by: $p(y_i|x_i, \mathbf{w}) = \mathcal{N}(\mathbf{w}^T X_i, \sigma^2)$

Because these are i.i.d. the probability of observing the dataset is

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \prod_{j=1}^{n_p} \mathcal{N}(\mathbf{w}^T X, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma_y^2}||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2\right)$$

This quantity is very important and is known as **likelihood**

Using the product of exponential and taking the logarithm we obtain the log-likelihood

$$\log(p(\mathbf{y}|\mathbf{x}, \mathbf{w})) = -n_p \log(\sqrt{2\pi}\sigma) - \frac{n_p}{2\sigma} J(\mathbf{w})$$

Thus, we see that the maximum of the log-likelihood is the maximum of the likelihood. Moreover, the maximum of the likelihood is attained at the minimum of our cost function!

The Analytical Solution

The solution to the optimization problem in the training of the linear regressor can be derived analytically.
It is easy to show that the cost function is a quadratic of the form:

$$J(\mathbf{w}) \propto \mathbf{r}^T\mathbf{r} = \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y}$$

Recalling the useful vector differentiation rules

$$\nabla_{\mathbf{x}}\Big(\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}\Big) = 2\mathbf{A}^T\mathbf{A}\mathbf{x} \quad \nabla_{\mathbf{x}}\Big(\mathbf{x}^T\mathbf{A}^T\mathbf{b}\Big) = \mathbf{A}^T\mathbf{b} \quad \nabla_{\mathbf{x}}\Big(\mathbf{b}^T\mathbf{A}\mathbf{x}\Big) = \mathbf{A}^T\mathbf{b}$$

We can see that the gradient w.r.t to the weights is $\boxed{\nabla_w J(\mathbf{w}) = \frac{2}{n_p}\mathbf{X}^T\Big(\mathbf{X}\mathbf{w} - \mathbf{y}\Big)}$

The Hessian is $\boxed{\mathcal{H}(\mathbf{w}) = \frac{1}{n_p}\mathbf{X}^T\mathbf{X}}$ A unique global minimum exist only if $\mathcal{H}(\mathbf{w}) > 0$

The Analytical Solution (2)

The minimum is achieved if (this result is known as normal equation)

$$\nabla_w J(\mathbf{w}) = 0 \rightarrow \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \rightarrow \boxed{\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

The inversion might be problematic if the Hessian has one eigenvalue equal (or close to) zero.

One possible solution is regularization: $\boxed{\mathbf{X}^T\mathbf{X} \leftarrow \mathbf{X}^T\mathbf{X} + \alpha\mathbf{I}}$

Identity matrix

$\alpha \geq 0, \alpha \in \mathbb{R}$

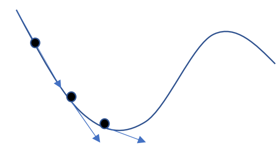
Penalty parameter

Will see later that such regularization is called Tikhonov regularization and leads to Ridge regression.

Table of Contents

von KARMAN INSTITUTE FOR FLUID DYNAMICS

A Numerical Solution



Most machine learning algorithm do not have an analytic solution of the minimization. Moreover, in many cases this is computationally prohibitive.

The classic solution is to use an iterative method and the simplest approach is the gradient descent:

$$\boxed{\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \underbrace{\eta(k)}_{\text{Learning rate, controlled by a learning 'schedule'}}\nabla_w J(\mathbf{w}; \mathbf{X}, \mathbf{y})}$$

$\mathbf{X} \in \mathbb{R}^{n_p \times 2}$

$\mathbf{y} \in \mathbb{R}^{n_p \times 1}$

Full data

A variant of this is the 'minibatch' gradient descent. This uses a (random) portion of the data in the update. The batch data can also be new data collected while training (online learning):

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta(k)\nabla_w J(\mathbf{w}; \mathbf{X}_b, \mathbf{y}_b)$$

$\mathbf{y}_b \in \mathbb{R}^{n_b \times 1}$

$\mathbf{X}_b \in \mathbb{R}^{n_b \times 2}$

Minibatch

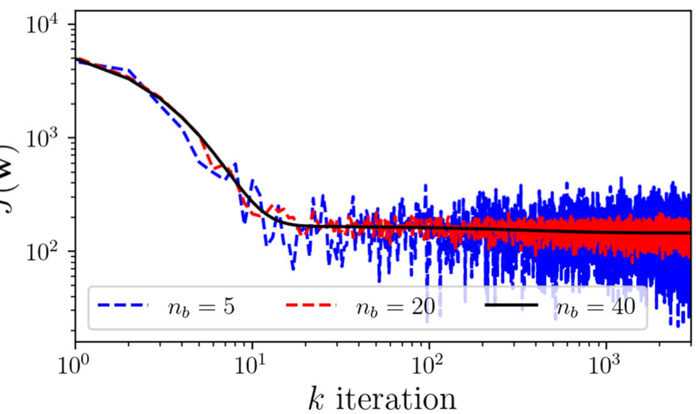
Definitions (user defined quantities):

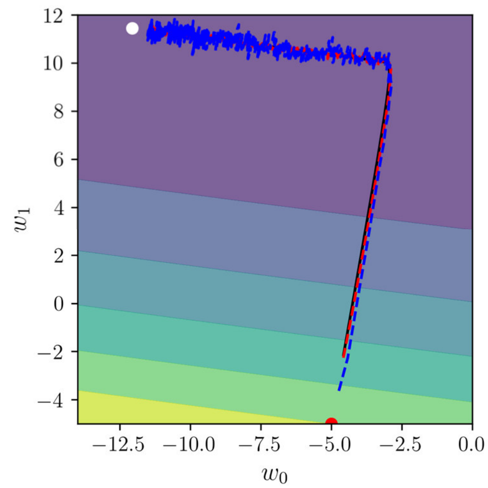
The number of samples in each portion is known as the **batch size.**

The number of 'complete passes' through the training data is known as **epochs.**

Example: is we have 1000 data points and work with batches of 100, then 1 epoch consists of 10 iterations

Evolution of the Learning Process





Slide 43

Table of Contents

Model Validation and Model Uncertainty

How much is the training depending on the specific data at hand? How sensitive are the weights to the data? How well is the model generalizing outside the training range?

To answer these questions, we need test (validation) data. We save a portion (usually 30%) of the data for validation and define two kinds of 'errors':

In-sample error

$$J_i(\mathbf{w}) = \frac{1}{n_*}||\mathbf{y}_* - \mathbf{X}_*\mathbf{w}||_2^2$$

Out of sample error

$$J_i(\mathbf{w}) = \frac{1}{n_{**}}||\mathbf{y}_{**} - \mathbf{X}_{**}\mathbf{w}||_2^2$$

The weights are computed by minimizing the in-sample error.

We could repeat this operation many times. Depending on how we do it, we call this cross-validation or Monte-Carlo or Ensemble validation:

1) **K-fold validation**: divide the data into K fold and repeat the training K times, each time using one-fold for validation

2) **Bootstrapping or Monte Carlo**: use a random sampling of the data (with or without resampling)

Model Validation and Model Uncertainty

Having repeated the previous validation multiple times (especially in the Monte-Carlo/Bootstrapping approach) we have a population of weights and many possible predictions. We can use the population to compute uncertainties in two ways:

1) **Based on the parameters (prior).** We might observe 'well-behaved' distributions on the weights. Then we can fit a distribution on the weights and see how this propagates to the prediction. We discuss this approach in Lecture 10, when we will use the 'affine transform' rule for Gaussians.

2) **Based on the ensemble statistics (posterior).** We could treat the population of prediction independently from the weights. Then our predictions and variance would be

$$\mu_y(x) = \frac{1}{n_e}\sum_{j=0}^{n_e-1} y_j(x) \qquad \sigma^2_{\bar{y}}(x) = \frac{1}{n_e}\sum_{j=0}^{n_e-1}\left(y_j(x) - \mu_y(x)\right)^2$$

Model Validation and Model Uncertainty

Let us compare the performances of two models for the available data. We use the ensemble statistics for the uncertainties.

1) A Linear Model:

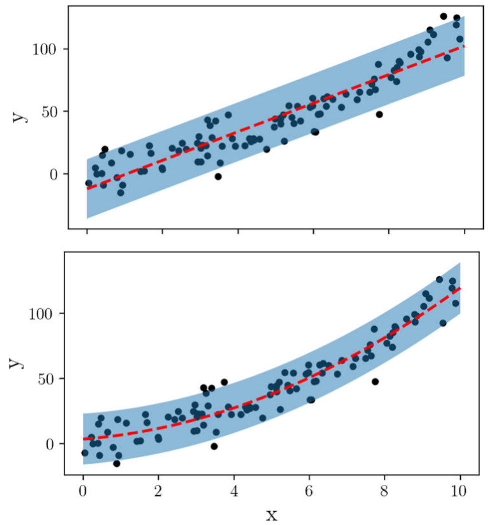
$$y = w_1 x + w_0$$

$$\overline{J}_i = 143.6 \quad \overline{J}_o = 155.2$$

2) Second Order Model:

$$y = w_2 x^2 + w_1 x + w_0$$

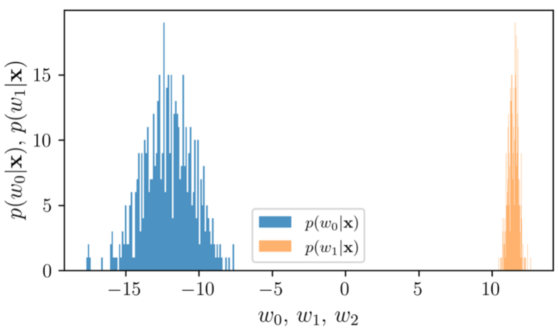
$$\overline{J}_i = 95.3 \quad \overline{J}_o = 106.0$$



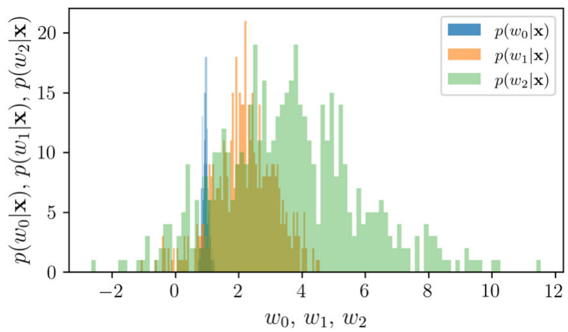
Estimating the weight variance

The results of the ensemble allows for analyzing the distribution (and the joint distributions!) of the model weights. If these follow 'classic' distributions, we could fit the right pdf and from there compute the uncertainty propagation analytically (approach 1)

First Order Model



Second Order Model:



Summary

1. Machine Learning Today

2. What is Machine Learning ?

3. Types of Machine Learning

4. Background for Exercise 1: Linear Regression

    4.1 Model Formulation

    4.2 The training problem and its analytical solution

    4.3 A numerical solution

    4.4 Model uncertainties

Make sure this is clear by Lecture 4 !

The end!

Slide 50