Statistical Machine Learning

Christian Walder

Machine Learning Research Group
CSIRO Data61

and

College of Engineering and Computer Science
The Australian National University

Canberra
Semester One, 2020.

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Outlines

Part II

Introduction

Flavour of this course

- Formalise intuitions about problems
- Use language of mathematics to express models
- Geometry, vectors, linear algebra for reasoning
- Probabilistic models to capture uncertainty
- Design and analysis of algorithms
- Numerical algorithms in python
- Understand the choices when designing machine learning methods

What is Machine Learning?

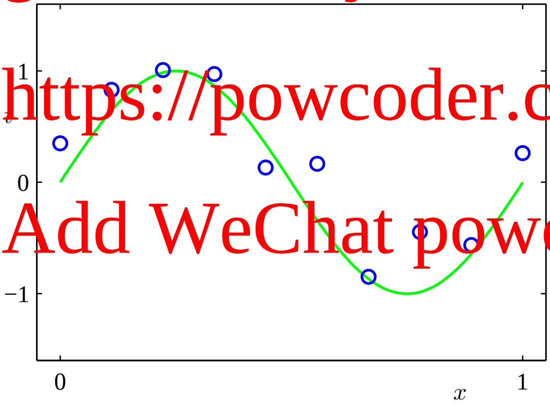
Definition (Mitchell, 1998)

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.

Polynomial Curve Fitting

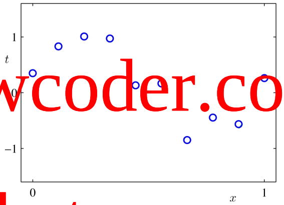
- some artificial data created from the function

$$\sin(2\pi x) + \text{random noise} \qquad x = 0, \dots, 1$$



Polynomial Curve Fitting - Input Specification

$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$



Polynomial Curve Fitting - Input Specification

$$N = 10$$

$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$

$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$

$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$

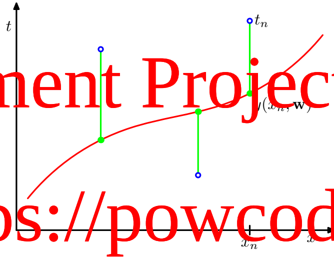
$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$

Polynomial Curve Fitting - Model Specification

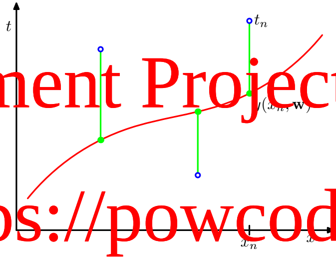
M : order of polynomial

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$
$$= \sum_{m=0}^{M} w_m x^m$$

- nonlinear function of x
- linear function of the unknown model parameter $\mathbf{w}$
- How can we find good parameters $\mathbf{w} = (w_1, \dots, w_M)^T$?


t
t_n
$y(x_n, \mathbf{w})$
x_n
x

Learning is Improving Performance



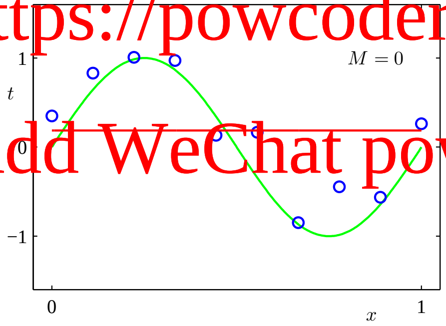
- Performance measure : Error between target and prediction of the model for the training data

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n\right)^2$$

- unique minimum of $E(\mathbf{w})$ for argument $\mathbf{w}^\star$ under certain conditions (what are they?)

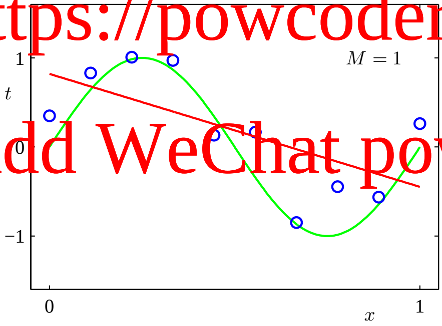
Model Comparison or Model Selection

$$y(x, \mathbf{w}) = \sum_{m=0}^{M} w_m x^m \Big|_{M=0} = w_0$$



Model Comparison or Model Selection

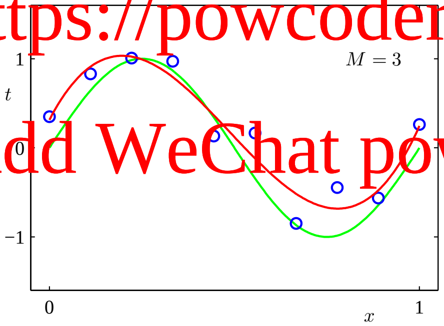
$$
\begin{aligned}
y(x, \mathbf{w}) &= \sum_{m=0}^{M} w_m x^m \Bigg|_{M=1} \\
&= w_0 + w_1 x
\end{aligned}
$$



Model Comparison or Model Selection

$$
y(x, \mathbf{w}) = \sum_{m=0}^{M} w_m x^m \Big|_{M=3}
$$

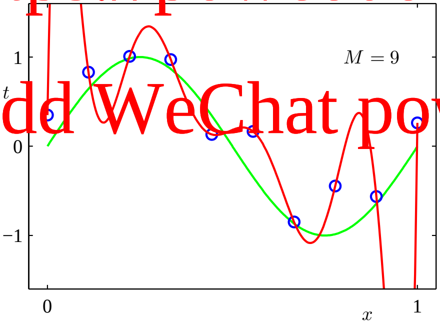
$$
= w_0 + w_1 x + w_2 x^2 + w_3 x^3
$$



Model Comparison or Model Selection

$$
\begin{aligned}
y(x, \mathbf{w}) &= \left. \sum_{m=0}^{M} w_m \, x^m \right|_{M=9} \\
&= w_0 + w_1 \, x + \cdots + w_8 \, x^8 + w_9 \, x^9
\end{aligned}
$$

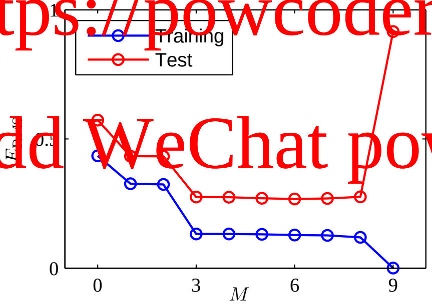
- over-fitting



Testing the Model

- Train the model and get $\mathbf{w}^\star$
- Get 100 new data points
- Root-mean-square (RMS) error

$$E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$$



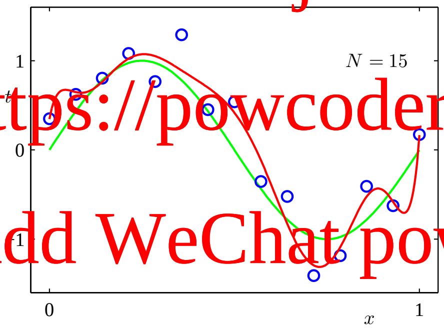
Testing the Model

	M = 0	M = 1	M = 3	M = 9
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Table: Coefficients $\mathbf{w}^\star$ for polynomials of various order.

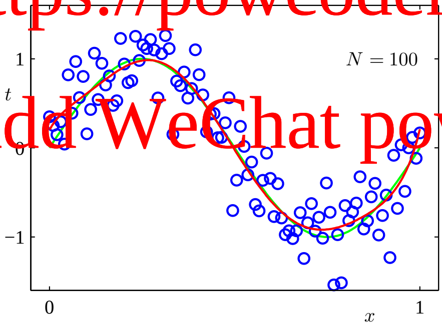
More Data

- $N = 15$



More Data

- $N = 100$
- heuristics : have no less than 5 to 10 times as many data points than parameters
- but number of parameters is not necessarily the most appropriate measure of model complexity !
- later: Bayesian approach



Regularisation

- How to constrain the growing of the coefficients $\mathbf{w}$?
- Add a regularisation term to the error function

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

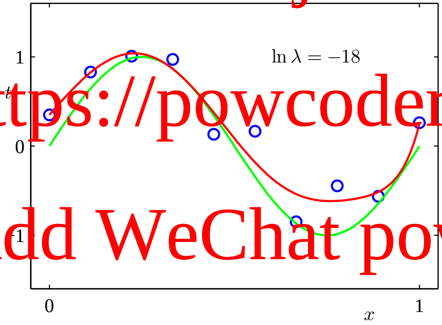
- Squared norm of the parameter vector $\mathbf{w}$

$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + \cdots + w_M^2$$

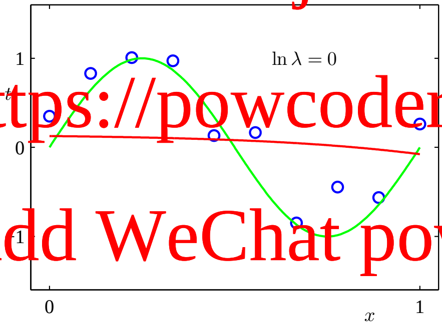
- unique minimum of $E(\mathbf{w})$ for argument $\mathbf{w}^\star$ under certain conditions (what are they for $\lambda = 0$? for $\lambda > 0$?)

Regularisation

- $M = 9$

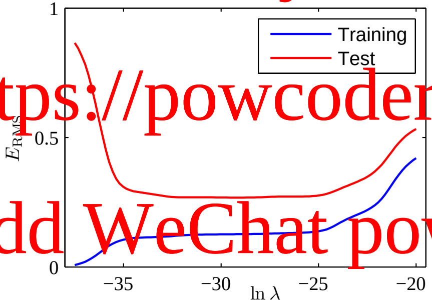


- $M = 9$



Regularisation

- $M = 9$

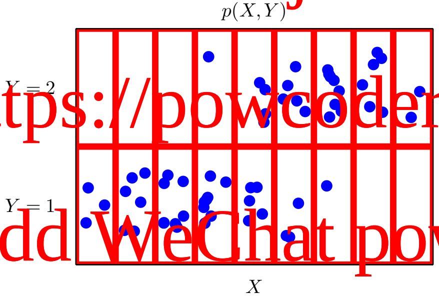


What is Machine Learning?

Definition (Mitchell, 1998)

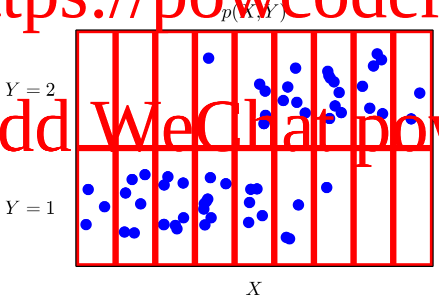
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

- Task: regression
- Experience: $\mathbf{x}$ input examples, $\mathbf{t}$ output labels
- Performance: squared error
- Model choice
- Regularisation
- **do not train on the test set!**


$p(X, Y)$
$Y = 2$
$Y = 1$
X

Probability Theory

Y vs. X	a	b	c	d	e	f	g	h	i	sum
2	0	0	0	1	4	5	8	6	2	26
1	3	6	8	8	5	3	1	0	0	34
sum	3	6	8	9	9	8	9	6	2	60



Sum Rule

Y vs. X	a	b	c	d	e	f	g	h	i	sum
2	0	0	0	1	4	5	8	6	2	26
1	3	6	8	8	5	3	1	0	0	34
sum	3	6	8	9	9	8	9	6	2	60

$$p(X = d, Y = 1) = 8/60$$

$$p(X = d) = p(X = d, Y = 2) + p(X = d, Y = 1)$$
$$= 1/60 + 8/60$$

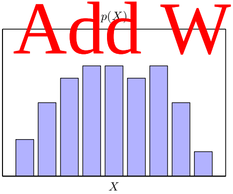
$$p(X = d) = \sum_Y p(X = d, Y)$$

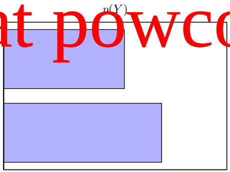
$$p(X) = \sum_Y p(X, Y)$$

Sum Rule

Y vs. X	a	b	c	d	e	f	g	h	i	sum
2	0	0	0	1	4	5	8	6	2	26
1	3	6	8	8	5	3	1	0	0	34
sum	3	6	8	9	9	8	9	6	2	60

$$p(X) = \sum_Y p(X, Y) \qquad p(Y) = \sum_X p(X, Y)$$





Product Rule

Y vs. X	a	b	c	d	e	f	g	h	i	sum
2	0	0	0	1	4	5	8	6	2	26
1	3	6	8	8	5	3	1	0	0	34
sum	3	6	8	9	9	8	9	6	2	60

Conditional Probability

$$p(X = d \mid Y = 1) = 8/34$$

Calculate $p(Y = 1)$:

$$p(Y = 1) = \sum_X p(X, Y = 1) = 34/60$$

$$p(X = d, Y = 1) = p(X = d \mid Y = 1)\, p(Y = 1)$$

$$p(X, Y) = p(X \mid Y)\, p(Y)$$

Another intuitive view is renormalisation of relative frequencies:

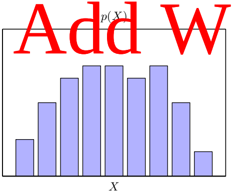
$$p(X \mid Y) = \frac{p(X, Y)}{p(Y)}$$

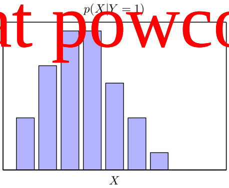
Sum and Product Rules

Y vs. X	a	b	c	d	e	f	g	h	i	sum
2	0	0	0	1	4	5	8	6	2	26
1	3	6	8	8	5	3	1	0	0	34
sum	3	6	8	9	9	8	9	6	2	60

$$p(X) = \sum_Y p(X, Y)$$

$$p(X \mid Y) = \frac{p(X, Y)}{p(Y)}$$





Sum Rule and Product Rule

- Sum Rule

$$p(X) = \sum_{Y} p(X, Y)$$

- Product Rule

$$p(X, Y) = p(X \mid Y)\, p(Y)$$

These rules form the basis of Bayesian machine learning, and this course!

Bayes Theorem

Use product rule

$$p(X, Y) = p(X \mid Y)\, p(Y) = p(Y \mid X)\, p(X)$$

Bayes Theorem

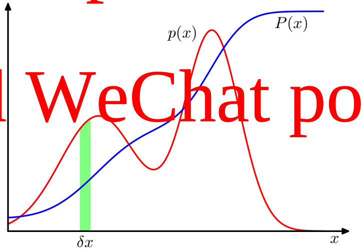
$$p(Y \mid X) = \frac{p(X \mid Y)\, p(Y)}{p(X)} \qquad \text{only defined for } p(X) > 0$$

and

$$\begin{aligned} p(X) &= \sum_Y p(X, Y) && \text{(sum rule)} \\ &= \sum_Y p(X \mid Y)\, p(Y) && \text{(product rule)} \end{aligned}$$

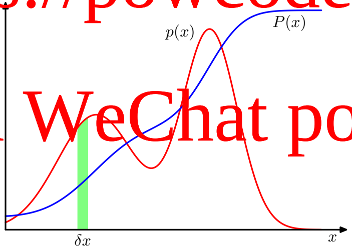
Probability Densities

- Real valued variable $x \in \mathbb{R}$
- Probability of x to fall in the interval $(x, x + \delta x)$ is given by $p(x)\delta x$ for infinitesimal small δx.
- $$p(x \in (a, b)) = \int_a^b p(x) \, \mathrm{d}x.$$



Constraints on $p(x)$

- Nonnegative
$$p(x) \geq 0$$
- Normalisation
$$\int_{-\infty}^{\infty} p(x) \, \mathrm{d}x = 1.$$

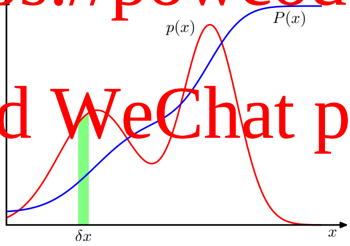


Cumulative distribution function $P(x)$

$$P(x) = \int_{-\infty}^{x} p(z)\, \mathrm{d}z$$

or

$$\frac{d}{dx} P(x) = p(x)$$



- Vector $\mathbf{x} \equiv (x_1, \dots, x_D)^T = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix}$

- Nonnegative

$$p(\mathbf{x}) \geq 0$$

- Normalisation

$$\int_{-\infty}^{\infty} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 1.$$

- This means

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(\mathbf{x}) \, \mathrm{d}x_1 \dots \, \mathrm{d}x_D = 1.$$

Sum and Product Rule for Probability Densities

- Sum Rule

$$p(x) = \int_{-\infty}^{\infty} p(x, y) \, \mathrm{d}y$$

- Product Rule

$$p(x, y) = p(y \mid x) \, p(x)$$

Expectations

- Weighted average of a function f(x) under the probability distribution $p(x)$

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \text{discrete distribution } p(x)$$

$$\mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x \qquad \text{probability density } p(x)$$

How to approximate $\mathbb{E}[f]$

- Given a finite number N of points x_n drawn from the probability distribution $p(x)$.
- Approximate the expectation by a finite sum:

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

- How to draw points from a probability distribution $p(x)$? Lecture coming about “Sampling”

Expection of a function of several variables

- arbitrary function $f(x, y)$

$$\mathbb{E}_x[f(x, y)] = \sum_x p(x) f(x, y) \qquad \text{discrete distribution } p(x)$$

$$\mathbb{E}_x[f(x, y)] = \int p(x) f(x, y) \, \mathrm{d}x \qquad \text{probability density } p(x)$$

- Note that $\mathbb{E}_x[f(x, y)]$ is a function of y.

Conditional Expectation

- arbitrary function $f(x)$

$$\mathbb{E}_x[f \mid y] = \sum_x p(x \mid y) f(x) \qquad \text{discrete distribution } p(x)$$

$$\mathbb{E}_x[f \mid y] = \int p(x \mid y) f(x)\, \mathrm{d}x \qquad \text{probability density } p(x)$$

- Note that $\mathbb{E}_x[f \mid y]$ is a function of y.
- Other notation used in the literature : $\mathbb{E}_{x|y}[f]$.
- What is $\mathbb{E}[\mathbb{E}[f(x) \mid y]]$? Can we simplify it?
- This must mean $\mathbb{E}_y[\mathbb{E}_x[f(x) \mid y]]$. (Why?)

$$\begin{aligned}
\mathbb{E}_y[\mathbb{E}_x[f(x) \mid y]] &= \sum_y p(y)\, \mathbb{E}_x[f \mid y] = \sum_y p(y) \sum_x p(x|y) f(x) \\
&= \sum_{x,y} f(x)\, p(x,y) = \sum_x f(x)\, p(x) \\
&= \mathbb{E}_x[f(x)]
\end{aligned}$$

Variance

- arbitrary function $f(x)$

$$\operatorname{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}\left[f(x)^2\right] - \mathbb{E}[f(x)]^2$$

- Special case: $f(x) = x$

$$\operatorname{var}[x] = \mathbb{E}\left[(x - \mathbb{E}[x])^2\right] = \mathbb{E}\left[x^2\right] - \mathbb{E}[x]^2$$

Covariance

- Two random variables $x \in \mathbb{R}$ and $y \in \mathbb{R}$

$$
\begin{aligned}
\operatorname{cov}[x, y] &= \mathbb{E}_{x,y}\left[(x - \mathbb{E}[x])(y - \mathbb{E}[y])\right] \\
&= \mathbb{E}_{x,y}[x\,y] - \mathbb{E}[x]\,\mathbb{E}[y]
\end{aligned}
$$

- With $\mathbb{E}[x] = a$ and $\mathbb{E}[y] = b$

$$
\begin{aligned}
\operatorname{cov}[x, y] &= \mathbb{E}_{x,y}\left[(x - a)(y - b)\right] \\
&= \mathbb{E}_{x,y}[x\,y] - \mathbb{E}_{x,y}[x\,b] - \mathbb{E}_{x,y}[a\,y] + \mathbb{E}_{x,y}[a\,b] \\
&= \mathbb{E}_{x,y}[x\,y] - b\,\underbrace{\mathbb{E}_{x,y}[x]}_{=\mathbb{E}_x[x]} - a\,\underbrace{\mathbb{E}_{x,y}[y]}_{=\mathbb{E}_y[y]} + a\,b\,\underbrace{\mathbb{E}_{x,y}[1]}_{=1} \\
&= \mathbb{E}_{x,y}[x\,y] - a\,b - a\,b + a\,b = \mathbb{E}_{x,y}[x\,y] - a\,b \\
&= \mathbb{E}_{x,y}[x\,y] - \mathbb{E}[x]\,\mathbb{E}[y]
\end{aligned}
$$

- Expresses how strongly x and y vary together. If x and y are independent, their covariance vanishes.

Covariance for Vector Valued Variables

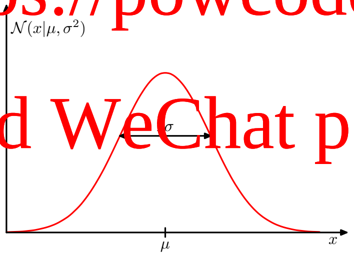
- Two random variables $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{y} \in \mathbb{R}^D$

$$\begin{aligned} \operatorname{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y}^{\mathbf{T}} - \mathbb{E}[\mathbf{y}^{\mathbf{T}}])\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\mathbf{x}\,\mathbf{y}^{\mathbf{T}}\right] - \mathbb{E}[\mathbf{x}]\,\mathbb{E}\left[\mathbf{y}^{\mathbf{T}}\right] \end{aligned}$$

The Gaussian Distribution

- $x \in \mathbb{R}$
- Gaussian Distribution with mean μ and variance σ^2

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\}$$



The Gaussian Distribution

- $\mathcal{N}(x \mid \mu, \sigma^2) > 0$
- $\int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2) \, dx = 1$
- Expectation over x

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2) \, x \, dx = \mu$$

- Expectation over x^2

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2) \, x^2 \, dx = \mu^2 + \sigma^2$$

- Variance of x

$$\operatorname{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Strategy in this course

- Estimate best predictor = training = learning
  Given data $(x_1, y_1), \dots, (x_n, y_n)$, find a predictor $f_{\mathbf{w}}(\cdot)$.

  1. Identify the type of input x and output y data
  2. Propose a (linear) mathematical model for $f_{\mathbf{w}}$
  3. Design an objective function or likelihood
  4. Calculate the optimal parameter ($\mathbf{w}$)
  5. Model uncertainty using the Bayesian approach
  6. Implement and compute (the algorithm in python)
  7. Interpret and diagnose results