Probabilistic machine learning

Karsten W. Jacobsen

Practical stuff

- Notes about probabilistic machine learning available at DTU Learn
- Jupyter/Python notebook available with code necessary to produce all figures - but try to make your own code!
- The lecture slides are also available, and will be updated as we go along
- Switching between lectures and exercises - we have to figure out what works the best
- Lectures:
  - Speak up to ask questions or
  - Use chat to ask questions. Andreas Vishart will keep an eye on the chat and alert me.
- Exercises:
  - Join in groups in “breakout rooms”
  - Use chat to get Andreas Vishart or me to join you and help
  - Use “blue hand” to signal that we can move on

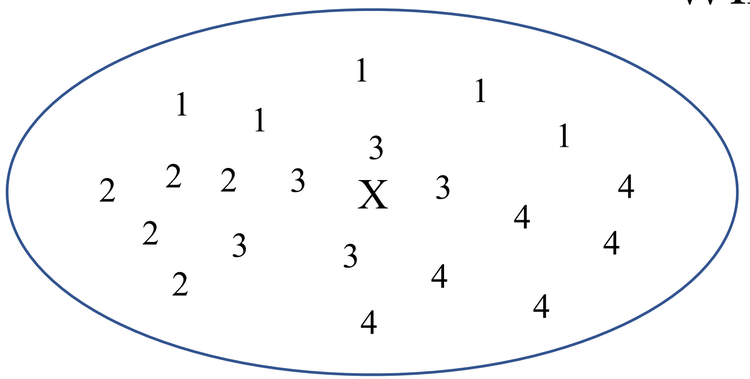
Probabilistic machine learning - overview

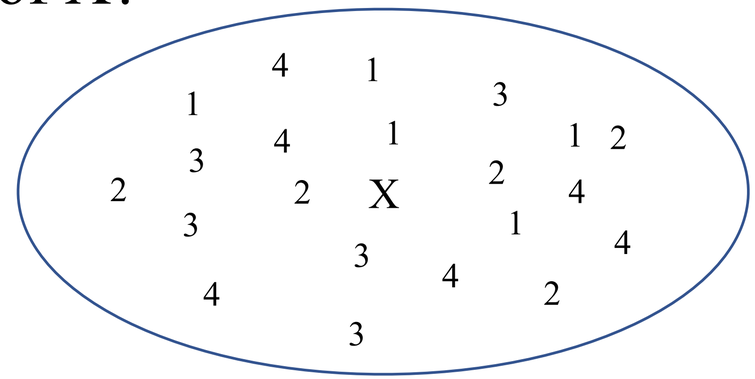
- Probability theory:
  - Intro, function fitting, probability theory, sum and product rules (pgs 1-14)
  - Bayes' theorem, models and data, model selection, probability distributions
  - Applications of Bayes' theorem
- Machine learning:
  - Methods of least squares, linear basis function models, regularized least squares, prior and posterior
  - Kernel regression and Gaussian processes
  - GP and hyperparameters
- Small project:
  - Perovskites

Machine learning

The kind of question we will ask:
I know the light absorption efficiency of 500 materials.
Now I discover a new material. Can I say anything about the light absorption of this material based on my previous knowledge?

What is the value of X?

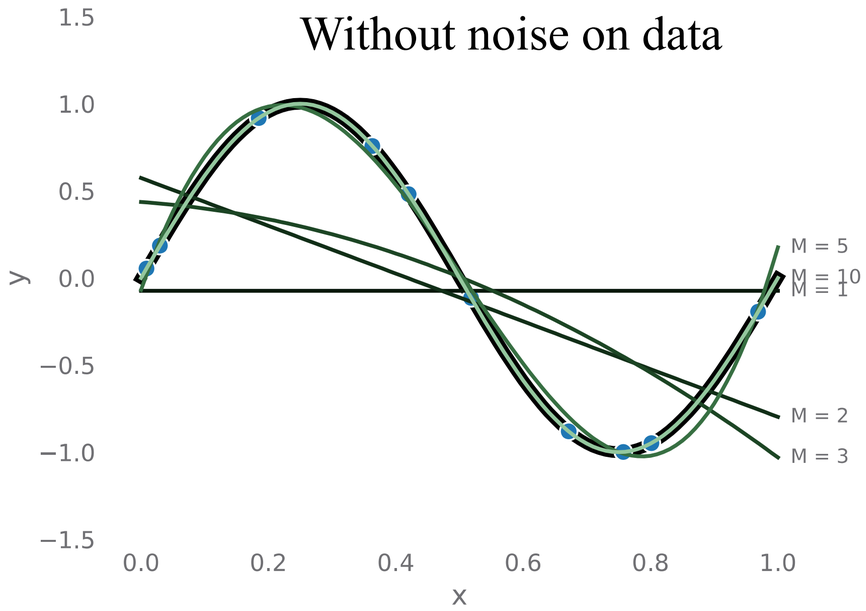


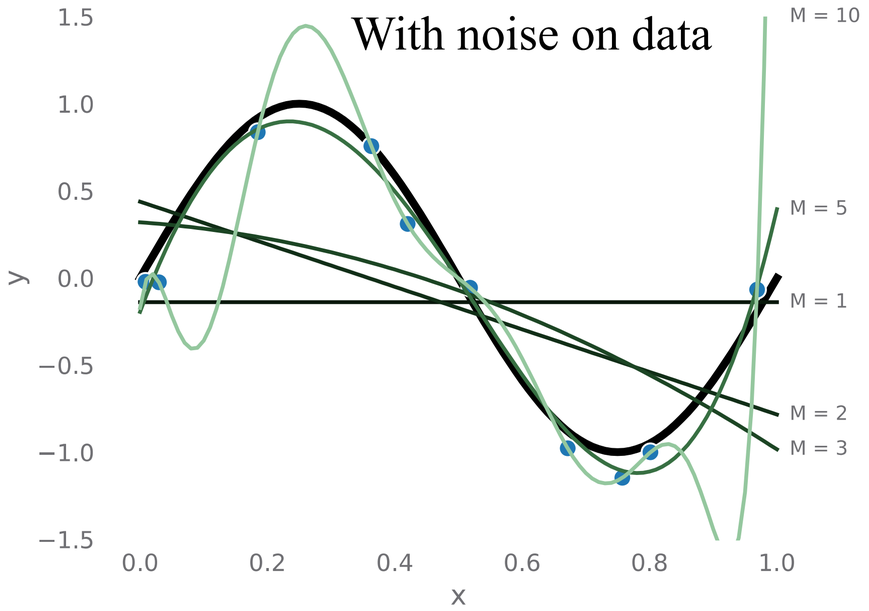


A key concept: correlation

The primary example: fitting a function

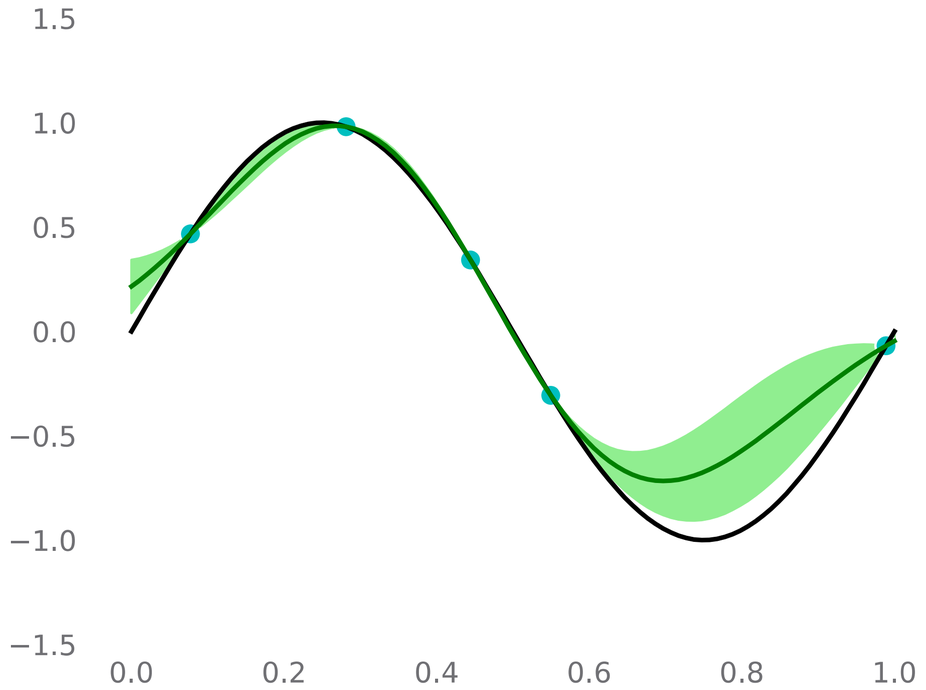
Polynomial fit: $f_{\text{fit}}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1}$





Too simple function (M too small): high *bias*
Too flexible function (M too high): high *variance* (*overfitting*)
What is the optimal value for M?

The goal:
Reliable predictions with uncertainty estimates



Prediction with *Gaussian Process*

Black curve: target function
Green curve: best prediction
Green area: uncertainty

To deal with uncertainties we need *probability theory.* Coming up next …

Probability theory

We assign probabilities to *propositions:*

Examples:

Y: When you through a die, it will show the number 3.
Y is an *event*, which can be investigated statistically (throw the die many times)

$$\mathcal{P}(Y) = 1/6$$

X: The best polynomial fit to the function is of order m=5
Not an event, but still a proposition.

$$\mathcal{P}(X) = ?$$

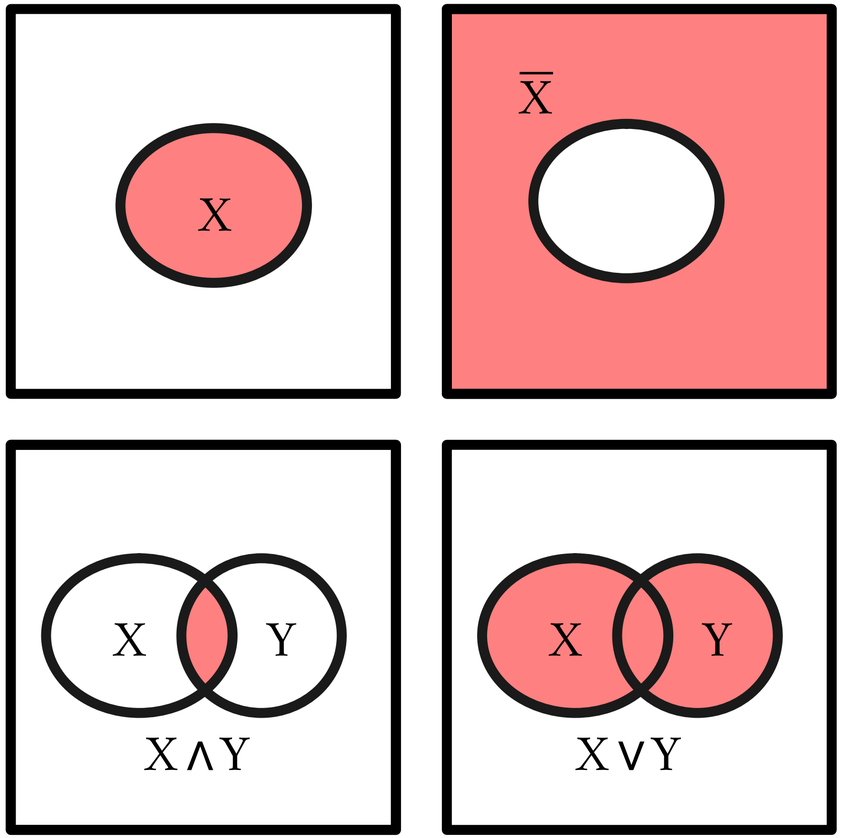
Probability theory: the ingredients

Not X : $\overline{X}$

X and Y : $X \wedge Y$ (or X, Y or just XY)

X or Y : $X \vee Y$ (or $X + Y$)

X	Y	$\overline{X}$	$X \wedge Y$	$X \vee Y$
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False



Probability theory: the ingredients

Not X : $\overline{X}$

X and Y : $X \wedge Y$ (or X, Y or just XY)

X or Y : $X \vee Y$ (or $X + Y$)

Example: Results of throwing a die

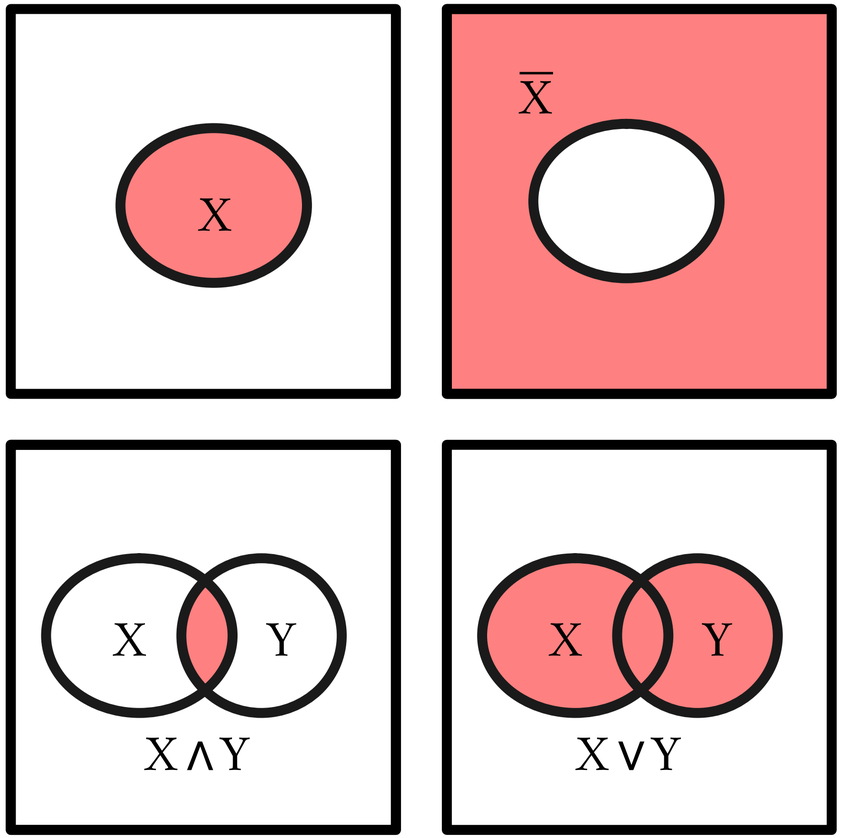
$X =$ the die shows 5 or 6 $= \{5, 6\}$

$Y = \{1, 6\}$

$\overline{X} = \{1, 2, 3, 4\}$

$X \wedge Y = \{6\}$

$X \vee Y = \{1, 5, 6\}$



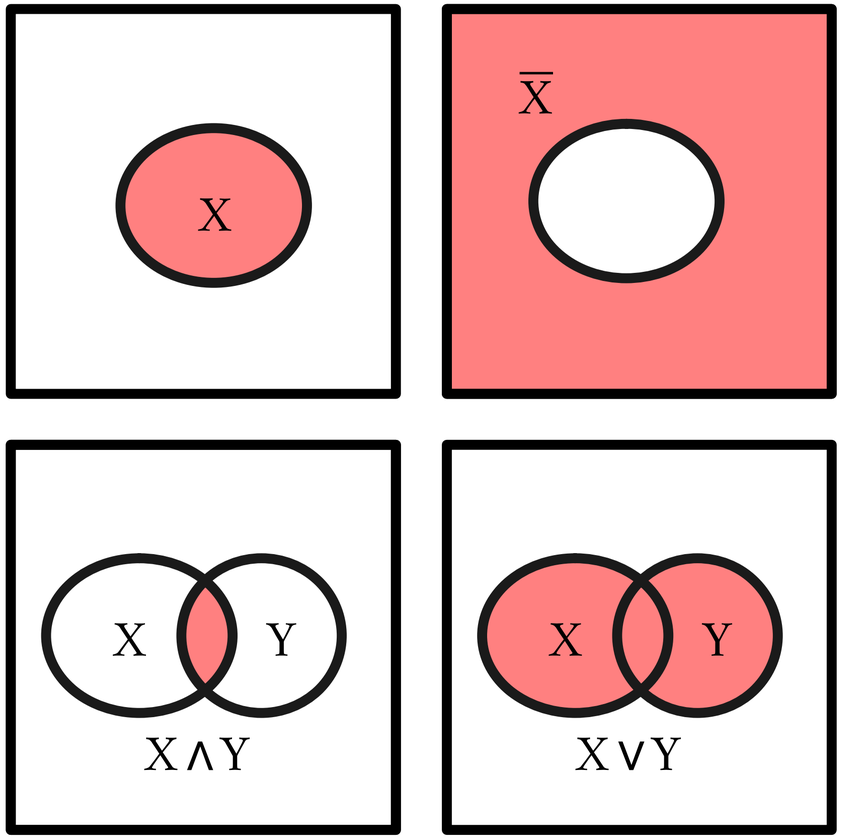
Probability theory: the ingredients

or using a truth table

Show graphically, that $X \vee Y = \overline{\overline{X} \wedge \overline{Y}}$

And that $\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$

And $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$



Form groups in the breakout rooms and discuss. Contact me or Andreas via chat if you have questions and we shall join the breakout room.

When you are done raise the blue hand in zoom!

Probability theory: the rules of the game

Positivity: $\mathcal{P}(X) \geq 0$

Sum rule: $\mathcal{P}(X) + \mathcal{P}(\overline{X}) = 1$ or $\mathcal{P}(X|I) + \mathcal{P}(\overline{X}|I) = 1$

Product rule: $\mathcal{P}(X,Y) = \mathcal{P}(X|Y)\mathcal{P}(Y)$ or $\mathcal{P}(X,Y|I) = \mathcal{P}(X|Y,I)\mathcal{P}(Y|I)$

Probability of X given I

Probability of X given Y

That is all!

The extended sum rule

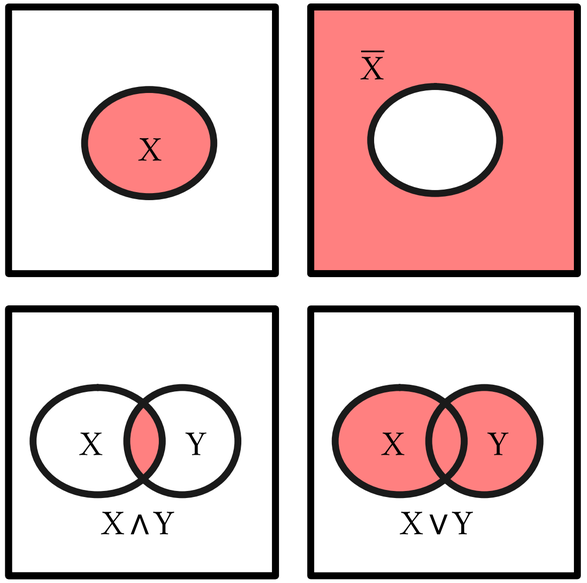
Sum rule: $\mathcal{P}(X) + \mathcal{P}(\overline{X}) = 1$

Product rule: $\mathcal{P}(X, Y) = \mathcal{P}(X|Y)\mathcal{P}(Y)$

$$\mathcal{P}(X, Y) = \mathcal{P}(X) + \mathcal{P}(Y) - \mathcal{P}(X \vee Y)$$

Proof:

$$\begin{aligned}
\mathcal{P}(X \vee Y) &= \mathcal{P}(\overline{\overline{X}\ \overline{Y}}) = 1 - \mathcal{P}(\overline{X}\ \overline{Y}) \\
&= 1 - \mathcal{P}(\overline{X}|\overline{Y})\mathcal{P}(\overline{Y}) = 1 - (1 - \mathcal{P}(X|\overline{Y}))\mathcal{P}(\overline{Y}) \\
&= 1 - \mathcal{P}(\overline{Y}) + \mathcal{P}(X\overline{Y}) = \mathcal{P}(Y) + \mathcal{P}(\overline{Y}|X)\mathcal{P}(X) \\
&= \mathcal{P}(Y) + (1 - \mathcal{P}(Y|X))\mathcal{P}(X) \\
&= \mathcal{P}(X) + \mathcal{P}(Y) - \mathcal{P}(XY)
\end{aligned}$$



Sum rule: $\mathcal{P}(X) + \mathcal{P}(\overline{X}) = 1$

Marginalization

Let Y_i be a set of mutually exclusive propositions ($\mathcal{P}(Y_i, Y_j) = 0$ for $i \neq j$), which together cover all possible propositions.

Then $\sum_{i=1}^{n} \mathcal{P}(Y_i) = 1$ or, if Y is continuous $\int \mathcal{P}(Y)\, dY = 1$

Using the product rule, we get

$$\mathcal{P}(X) = \sum_{i=1}^{n} \mathcal{P}(Y_i|X)\mathcal{P}(X) = \sum_{i=1}^{n} \mathcal{P}(Y_i, X) = \sum_{i=1}^{n} \mathcal{P}(X|Y_i)\mathcal{P}(Y_i) = \left(\int \mathcal{P}(X|Y)\mathcal{P}(Y)\, dY \right)$$

The process of "integrating out" Y is called *marginalization*.

Independence

A proposition X is said to be *independent* of another proposition Y if

$$\mathcal{P}(X|Y) = \mathcal{P}(X)$$

i.e. the probability of X is unchanged by Y being true or not.

For independent (also called *uncorrelated*) propositions we have

$$\mathcal{P}(X, Y) = \mathcal{P}(X|Y)\mathcal{P}(Y) = \mathcal{P}(X)\mathcal{P}(Y)$$

Exercises

- Use the sum and product rules to solve the exercises in the notes about
  - The product rule
  - The extended sum rule (I just did that)
  - The extended, extended sum rule
  - Independence of propositions
- Come back to main room at 11:45

Bayes' theorem

Reshuffle the equation $\mathcal{P}(X,Y) = \mathcal{P}(X|Y)\mathcal{P}(Y) = \mathcal{P}(Y|X)\mathcal{P}(X)$

and we get *Bayes' theorem*

$$\mathcal{P}(Y|X) = \frac{\mathcal{P}(X|Y)\mathcal{P}(Y)}{\mathcal{P}(X)} \qquad \left(\text{or } \mathcal{P}(Y|X,I) = \frac{\mathcal{P}(X|Y,I)\mathcal{P}(Y|I)}{\mathcal{P}(X|I)}\right)$$

This is the most important formula in this course!

Bayes' theorem and rational reasoning

From the book by Jaynes:
Suppose some dark night a policeman walks down a street, apparently deserted. Suddenly he hears a burglar alarm, looks across the street, and sees a jewelry store with a broken window. Then a gentleman wearing a mask comes crawling out through the broken window, carrying a bag which turns out to be full of expensive jewelry. The policeman doesn't hesitate at all in deciding that this gentleman is dishonest. But by what reasoning process does he arrive at this conclusion?

$$\mathcal{P}(\text{thief}|\text{behavior}) = \frac{\mathcal{P}(\text{behavior}|\text{thief})\mathcal{P}(\text{thief})}{\mathcal{P}(\text{behavior})}$$

Example: Alice, Bob, and their two kids

Alice and Bob have two children. What is the probability that they have two girls when you get the following information (four different cases)?

They have at least one girl (“≥1g”)
The oldest child is a girl (“old”)
They have a girl with blue eyes (“blue”)
They have a girl with the name “Mushroom” (“Mushroom”)

They have at least one girl

$$\mathcal{P}(2\text{g}|\geq 1\text{g}) = \frac{\mathcal{P}(\geq 1\text{g}|2\text{g})\mathcal{P}(2\text{g})}{\mathcal{P}(\geq 1\text{g})} = \frac{\mathcal{P}(2\text{g})}{\mathcal{P}(\geq 1\text{g})} = \frac{1/4}{3/4} = \frac{1}{3}$$

The oldest child is a girl

$$\mathcal{P}(2\text{g}|\text{old}) = \frac{\mathcal{P}(\text{old}|2\text{g})\mathcal{P}(2\text{g})}{\mathcal{P}(\text{old})} = \frac{\mathcal{P}(2\text{g})}{\mathcal{P}(\text{old})} = \frac{1/4}{1/2} = \frac{1}{2}$$

They have a girl with blue eyes
(probability of blue eyes: p)

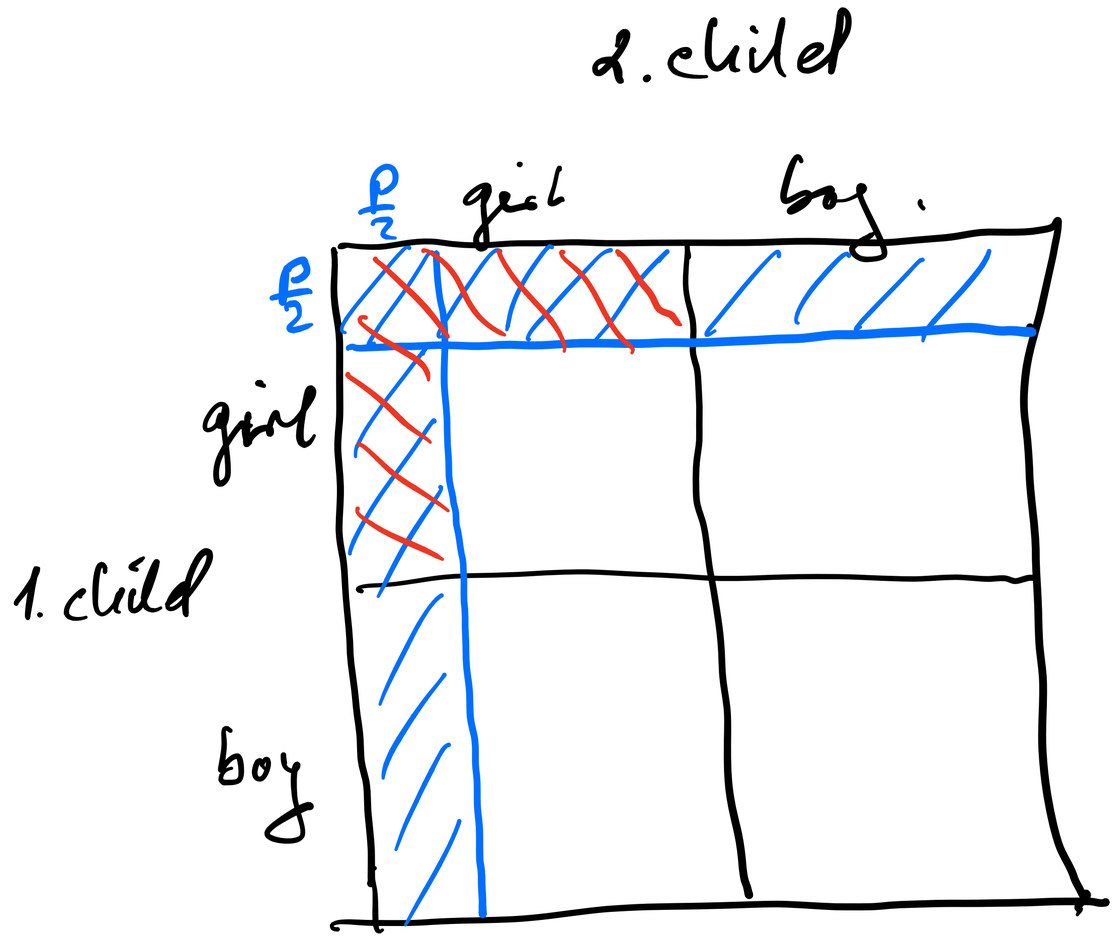
$$\mathcal{P}(2\text{g}|\text{blue}) = \frac{\mathcal{P}(\text{blue}|2\text{g})\mathcal{P}(2\text{g})}{\mathcal{P}(\text{blue})} = \frac{\mathcal{P}(\text{blue}|2\text{g})\mathcal{P}(2\text{g})}{\mathcal{P}(\text{blue}|2\text{g})\mathcal{P}(2\text{g}) + \mathcal{P}(\text{blue}|1\text{g})\mathcal{P}(1\text{g})}$$

$$= \frac{\left(p^2 + 2p(1-p)\right)\cdot 1/4}{\left(p^2 + 2p(1-p)\right)\cdot 1/4 + p\cdot 1/2} = \frac{2-p}{4-p}$$

$$\text{So if } p \sim 1/2, \text{ we get } \mathcal{P}(2\text{g}|\text{blue}) \sim \frac{3}{7}$$

They have a girl with the name “Mushroom”
(Same analysis as for the blue eyes, but now $p \ll 1$)

$$\mathcal{P}(2\text{g}|\text{Mushroom}) \sim \frac{1}{2}$$



$$P(2g \mid g \text{ with property } p) = \frac{\text{red area}}{\text{blue area}} = \frac{2 \cdot \frac{p}{2} \cdot \frac{1}{2} - \left(\frac{p}{2}\right)^2}{2 \cdot \frac{p}{2} - \left(\frac{p}{2}\right)^2} = \frac{2-p}{4-p}$$

Exercise

- Now do the exercise on medical screening
- Let us meet in 15 minutes

Medical screening

$$\mathcal{P}(ill|pos) = \frac{\mathcal{P}(pos|ill)\mathcal{P}(ill)}{\mathcal{P}(pos)} = \frac{\mathcal{P}(pos|ill)\mathcal{P}(ill)}{\mathcal{P}(pos|ill)\mathcal{P}(ill) + \mathcal{P}(pos|h)\mathcal{P}(h)}$$

$$= \frac{0.99 \cdot 0.001}{0.99 \cdot 0.001 + 0.01 \cdot 0.999} \sim 0.09$$

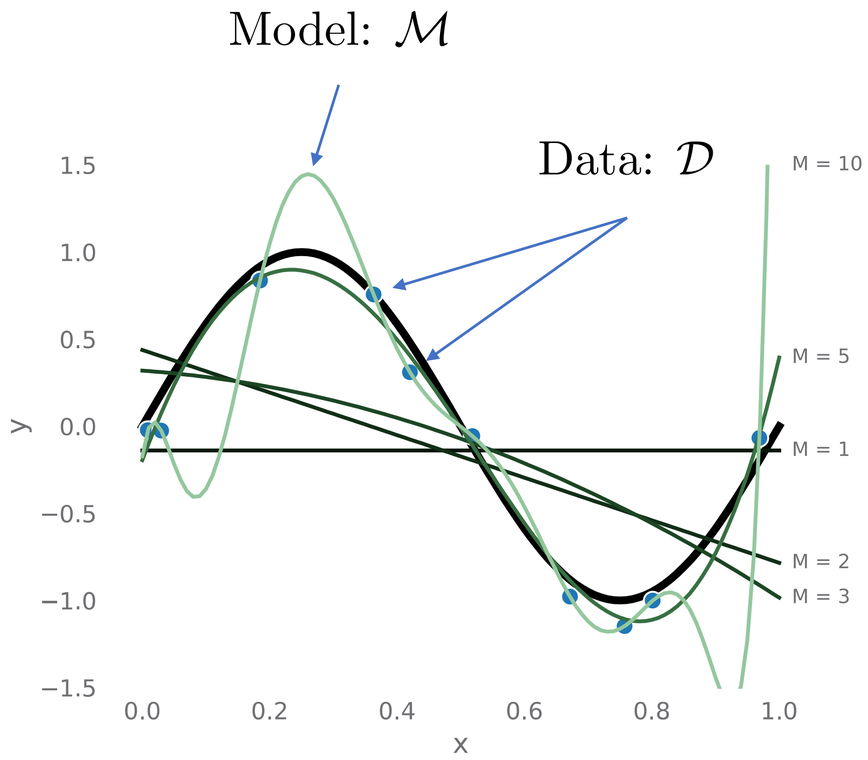
Bayes' theorem: models and data

$$\mathcal{P}(\mathcal{M}|\mathcal{D}) = \frac{1}{\mathcal{P}(\mathcal{D})}\mathcal{P}(\mathcal{D}|\mathcal{M})\mathcal{P}(\mathcal{M})$$

Posterior probability distribution

Likelihood

Prior probability distribution



Normalization:

$$\mathcal{P}(\mathcal{D}) = \mathcal{P}(\mathcal{D},\mathcal{M}) + \mathcal{P}(\mathcal{D},\bar{\mathcal{M}}) = \mathcal{P}(\mathcal{D}|\mathcal{M})\mathcal{P}(\mathcal{M}) + \mathcal{P}(\mathcal{D}|\bar{\mathcal{M}})\mathcal{P}(\bar{\mathcal{M}})$$

Why do we believe Newtons's second law?

Bayes' theorem - updating with new data

$$\mathcal{P}(\mathcal{M}|\mathcal{D}_{new},\mathcal{D}_{old}) = \frac{1}{\mathcal{P}(\mathcal{D}_{new}|\mathcal{D}_{old})}\mathcal{P}(\mathcal{D}_{new}|\mathcal{M},\mathcal{D}_{old})\mathcal{P}(\mathcal{M}|\mathcal{D}_{old})$$

New posterior distribution

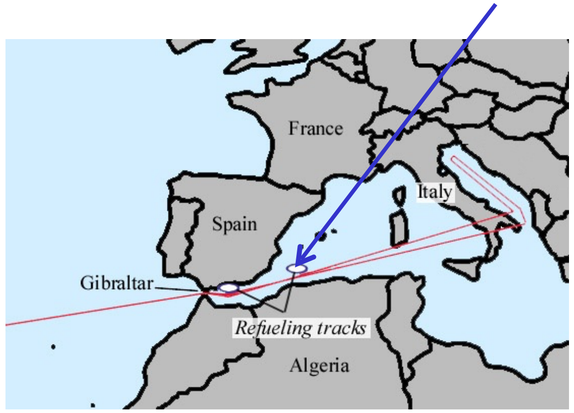
Likelihood of new data

Prior based on old data

Bayesian Search Theory in Practice
The 1966 Palomares B-52 crash

B-52G collided with KC-135 tanker when fueling

Refueling



4 H-bombs dropped
3 on land
1 in the Mediterranean Sea

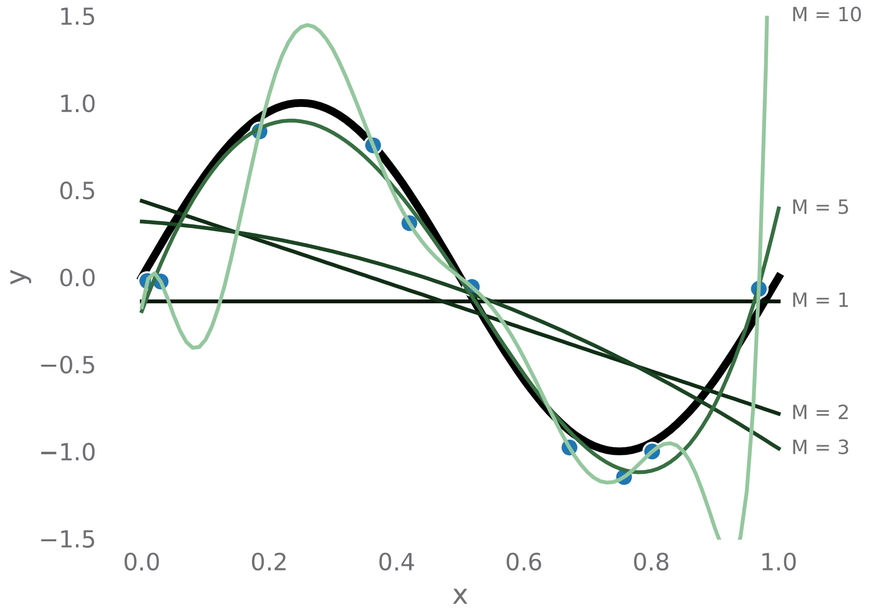
Bayesian search theory applied:
Assign probabilities to different areas of the sea
based on available information
(a local fisherman saw the bomb dropping)
Update your probability depending on your search.

Bomb recovered

Parameter estimation

$$f_{\mathrm{fit}}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1}$$

$$\begin{aligned}
\mathcal{P}(\boldsymbol{w}|\mathcal{D},\mathcal{M}) &= \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w},\mathcal{M})}{\mathcal{P}(\mathcal{D},\mathcal{M})} = \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w},\mathcal{M})}{\int \mathcal{P}(\mathcal{D},\boldsymbol{w}',\mathcal{M})d\boldsymbol{w}'} \\
&= \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w},\mathcal{M})}{\int \mathcal{P}(\mathcal{D}|\boldsymbol{w}',\mathcal{M})\mathcal{P}(\boldsymbol{w}',\mathcal{M})d\boldsymbol{w}'} \\
&= \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w}|\mathcal{M})\mathcal{P}(\mathcal{M})}{\int \mathcal{P}(\mathcal{D}|\boldsymbol{w}',\mathcal{M})\mathcal{P}(\boldsymbol{w}'|\mathcal{M})d\boldsymbol{w}'\mathcal{P}(\mathcal{M})} \\
&= \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w}|\mathcal{M})}{\int \mathcal{P}(\mathcal{D}|\boldsymbol{w}',\mathcal{M})\mathcal{P}(\boldsymbol{w}'|\mathcal{M})d\boldsymbol{w}'}
\end{aligned}$$



The denominator just ensures $\int \mathcal{P}(\boldsymbol{w}|\mathcal{D},\mathcal{M})d\boldsymbol{w} = 1$

Librarian or farmer? (Tversky and Kahneman)

An individual has been described by a neighbor as follows: “Steve is very shy and withdrawn, invariably helpful but with very little interest in people or in the world of reality. A meek and tidy soul, he has a need for order and structure, and a passion for detail.” Is Steve more likely to be a librarian or a farmer?

$$\frac{\mathcal{P}(\text{farmer}|\text{description})}{\mathcal{P}(\text{librarian}|\text{description})} = \frac{\mathcal{P}(\text{description}|\text{farmer})}{\mathcal{P}(\text{description}|\text{librarian})} \frac{\mathcal{P}(\text{farmer})}{\mathcal{P}(\text{librarian})}$$

In Denmark there are ~10000 farms, but only ~250 public libraries

Model selection

Comparing two models

$$\frac{\mathcal{P}(\mathcal{M}_2|\mathcal{D})}{\mathcal{P}(\mathcal{M}_1|\mathcal{D})} = \frac{\mathcal{P}(\mathcal{D}|\mathcal{M}_2)}{\mathcal{P}(\mathcal{D}|\mathcal{M}_1)} \cdot \frac{\mathcal{P}(\mathcal{M}_2)}{\mathcal{P}(\mathcal{M}_1)}$$

$$= \frac{\int \mathcal{P}(\mathcal{D}|\mathcal{M}_2, \boldsymbol{w}_2)\mathcal{P}(\boldsymbol{w}_2|\mathcal{M}_2)d\boldsymbol{w}_2}{\int \mathcal{P}(\mathcal{D}|\mathcal{M}_1, \boldsymbol{w}_1)\mathcal{P}(\boldsymbol{w}_1|\mathcal{M}_1)d\boldsymbol{w}_1} \cdot \frac{\mathcal{P}(\mathcal{M}_2)}{\mathcal{P}(\mathcal{M}_1)}$$

Probability distributions - definitions

Probability distribution $\mathcal{P}(x)$ Normalization: $\int \mathcal{P}(x)\,dx = 1$

Average value of a function: $\langle f \rangle = \int f(x)\mathcal{P}(x)\,dx,$

Variance: $\mathrm{Var}[f] = \langle (f - \langle f \rangle)^2 \rangle = \langle f^2 \rangle - \langle f \rangle^2$ In particular: $\mathrm{Var}[x] = \langle x^2 \rangle - \langle x \rangle^2$

Several variables: $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$ $\boldsymbol{x}\boldsymbol{x}^T = \begin{pmatrix} x_1x_1 & x_1x_2 & \cdots & x_1x_D \\ x_2x_1 & x_2x_2 & \cdots & x_2x_D \\ \vdots & \vdots & & \vdots \\ x_Dx_1 & x_Dx_2 & \cdots & x_Dx_D \end{pmatrix}$

Covariance matrix: $\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{x}^T] = \langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle)(\boldsymbol{x}^T - \langle \boldsymbol{x}^T \rangle) \rangle = \langle \boldsymbol{x}\boldsymbol{x}^T \rangle - \langle \boldsymbol{x} \rangle \langle \boldsymbol{x} \rangle^T$

If x_i and x_j are independent: $\mathrm{Cov}[x_i, x_j] = 0$

The normal (Gaussian) distribution

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(x-\mu)^2/2\sigma^2}$$

$$\langle x \rangle = \mu$$

$$\mathrm{Var}[x] = \sigma^2$$

Many nice mathematical properties. A product of two Gaussians is for example a new Gaussian:

$$\mathcal{N}(x|\mu,\sigma^2) \propto \mathcal{N}(x|\mu_1,\sigma_1^2) \cdot \mathcal{N}(x|\mu_2,\sigma_2^2) \qquad \text{with} \qquad \mu = \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right)\sigma^2 \quad \text{and} \quad \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

The central limit theorem (vaguely formulated): If a stochastic variable y is a sum of many stochastic variables $y = y_1 + y_2 + \ldots + y_N$ then as $N \to \infty$ the distribution $\mathcal{P}(y)$ becomes a normal distribution.

Binomial distribution

Consider a coin with probability *p* to get heads. The probability of getting *n* heads, when you throw the coin *N* times is

$$\mathcal{P}(n|N,p) = \binom{N}{n} p^n (1-p)^{(N-n)}$$

$$\langle n \rangle = pN$$

$$\mathrm{Var}[n] = Np(1-p)$$

For large *N* we get the normal distribution (to be used in an exercise):

$$\mathcal{P}(n|N,p) \sim \frac{1}{N}\mathcal{N}(x = n/N|p, p(1-p)/N)$$

Exercises about Bayes' theorem

- Now do the exercises on Bayes' theorem:
  - Gaussianly distributed data
  - Tversky and Kahneman
  - Playing with priors - the average height of Norwegians
  - Is the coin fair?
  - (But is the coin really fair?)

Exercise: Gaussianly distributed data

Data $x_1, x_2, \ldots x_N$ which we assume to be Gaussianly distributed with known variance σ^2. What is the best estimate of the mean of the data?

$$\begin{aligned}
\mathcal{P}(\mu|x_1, x_2, \ldots x_N, \sigma^2) &\propto \mathcal{P}(x_1, x_2, \ldots x_N|\mu, \sigma^2)\mathcal{P}(\mu|\sigma^2) \\
&\propto \mathcal{P}(x_1, x_2, \ldots x_N|\mu, \sigma^2) \\
&\propto \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \sigma^2) \propto \prod_{i=1}^{N} e^{-(x_i-\mu)^2/2\sigma^2} \\
&\propto \exp\left(-\sum_{i=1}^{N}(x_i-\mu)^2/2\sigma^2\right) \\
&\propto \exp\left(-((\mu-\bar{x})^2 + \overline{\Delta x^2})/2\sigma_N^2\right) \\
&\propto \exp\left(-(\mu-\bar{x})^2/2\sigma_N^2\right)
\end{aligned}$$

$$\bar{x} = \frac{1}{N}\sum_i x_i$$

$$\overline{\Delta x^2} = \frac{1}{N}\sum_i (x_i - \bar{x})^2$$

$$\sigma_N = \sigma/\sqrt{N}$$

The highest probability is obtained with

$$\mu = \bar{x}$$

and the standard deviation (i.e. the square root of the variance) is

$$\sigma/\sqrt{N}$$

Average height of Norwegians

$$\mathcal{P}(h_{\mathrm{Nor}}|h_1, h_2 \ldots h_N) \propto \mathcal{P}(h_1, h_2 \ldots h_N|h_{\mathrm{Nor}})\mathcal{P}(h_{\mathrm{Nor}})$$

$$\propto \left(\prod_n \mathcal{N}(h_n|h_{\mathrm{Nor}}, \sigma^2_{\mathrm{Nor}})\right) \mathcal{N}(h_{\mathrm{Nor}}|h_{\mathrm{DK}}, \sigma^2_d)$$

$$\propto \mathcal{N}(h_{\mathrm{Nor}}|\bar{h}, \sigma^2_{\mathrm{Nor}}/N)\ \mathcal{N}(h_{\mathrm{Nor}}|h_{\mathrm{DK}}, \sigma^2_d)$$

$$\propto \mathcal{N}(h_{\mathrm{Nor}}|h_0, \sigma^2_0)$$

$$\bar{h} = \frac{1}{N}\sum_{i=1}^{N} h_i$$

$$h_0 = \frac{\frac{\bar{h}}{\sigma^2_{\mathrm{Nor}}/N} + \frac{h_{\mathrm{DK}}}{\sigma^2_d}}{N/\sigma^2_{\mathrm{Nor}} + 1/\sigma^2_d}$$

$$= \frac{\sum_i h_i + (\frac{\sigma_{\mathrm{Nor}}}{\sigma_d})^2 h_{\mathrm{DK}}}{N + (\frac{\sigma_{\mathrm{Nor}}}{\sigma_d})^2}$$

$$\frac{\sigma^2_{\mathrm{Nor}}}{\sigma^2_d} \sim \frac{(5\,\mathrm{cm})^2}{(2.5\,\mathrm{cm})^2} \sim 4$$

“Pseudo-count” of 4 Danes.

$$\sigma^2_0 = \frac{\sigma^2_{\mathrm{Nor}}}{N + (\frac{\sigma_{\mathrm{Nor}}}{\sigma_d})^2}$$

Is the coin fair?

You get N_h heads out of N tosses. Is the coin fair?

$$\mathcal{P}(p|N_h, N) \propto \mathcal{P}(N_h|p, N)\mathcal{P}(p) \propto \mathcal{P}(N_h|p, N)$$
$$= \binom{N}{N_h} p^{N_h}(1-p)^{N_t} \propto \mathcal{B}(p|\alpha = N_h + 1, \beta = N_t + 1)$$

Probability to get heads: p
Tails: $N_t = N - N_h$

The best estimate (the average) of p:

$$\langle p \rangle = \frac{N_h + 1}{(N_h + 1) + (N_t + 1)} = \frac{N_h + 1}{N + 2}$$

Pseudo-count of 1 head and 1 tail

$$\sigma(p) = \sqrt{\mathrm{Var}[p]} = \sqrt{\frac{(N_h + 1)(N_t + 1)}{(N + 2)^2(N + 3)}}$$

Examples:

6 heads out of 10: $\langle p \rangle \pm \sigma(p) = 0.58 \pm 0.14$ Could be fair.

600 heads out of 1000: $\langle p \rangle \pm \sigma(p) = 0.60 \pm 0.02$ Is probably not fair

Beta distribution:

$$\mathcal{B}(x|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

with $$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

$$\langle x \rangle = \frac{\alpha}{\alpha + \beta}$$

$$\mathrm{Var}[x] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

Is the coin fair?

You get N_h heads out of N tosses. Is the coin fair?

$$\mathcal{P}(p|N_h, N) \propto \mathcal{P}(N_h|p, N)\mathcal{P}(p) \propto \mathcal{P}(N_h|p, N)$$
$$= \binom{N}{N_h} p^{N_h}(1-p)^{N_t} \propto \mathcal{B}(p|\alpha = N_h + 1, \beta = N_t + 1)$$

The best estimate (the average) of p:

$$\langle p \rangle = \frac{N_h + 1}{(N_h + 1) + (N_t + 1)} = \frac{N_h + 1}{N + 2}$$

Pseudo-count of 1 head and 1 tail

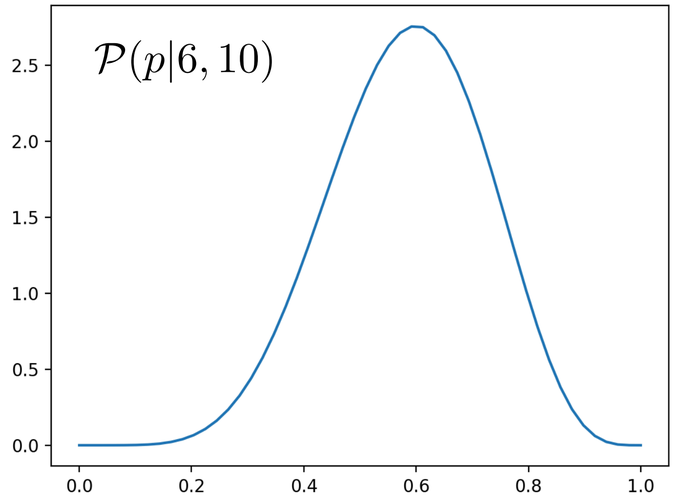
$$\sigma(p) = \sqrt{\mathrm{Var}[p]} = \sqrt{\frac{(N_h + 1)(N_t + 1)}{(N + 2)^2(N + 3)}}$$

Examples:

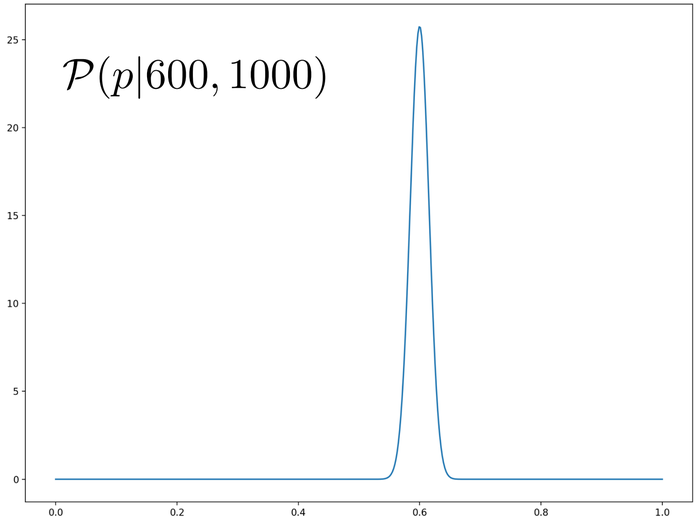
6 heads out of 10: $\langle p \rangle \pm \sigma(p) = 0.58 \pm 0.14$ Could be fair.

600 heads out of 1000: $\langle p \rangle \pm \sigma(p) = 0.60 \pm 0.02$ Is probably not fair

6 heads out of 10

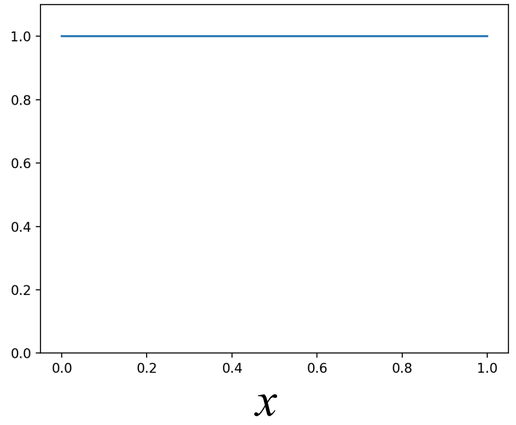


600 heads out of 1000

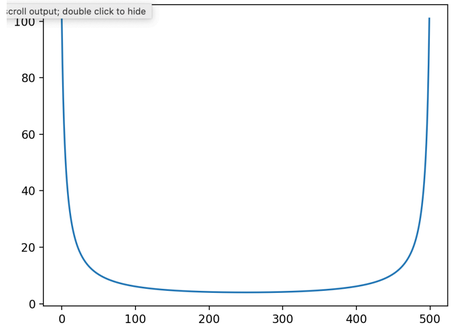


Is the coin fair? - changing the prior

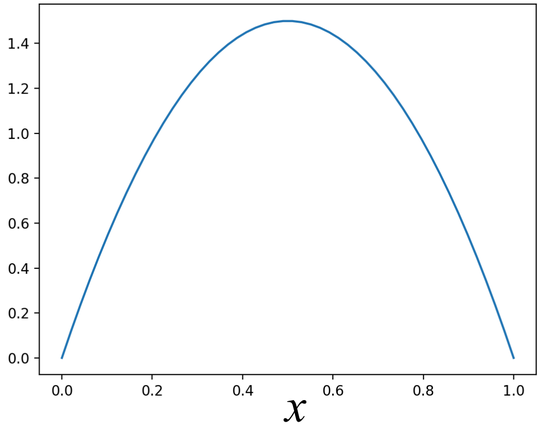
Constant prior: $\alpha = \beta = 1$



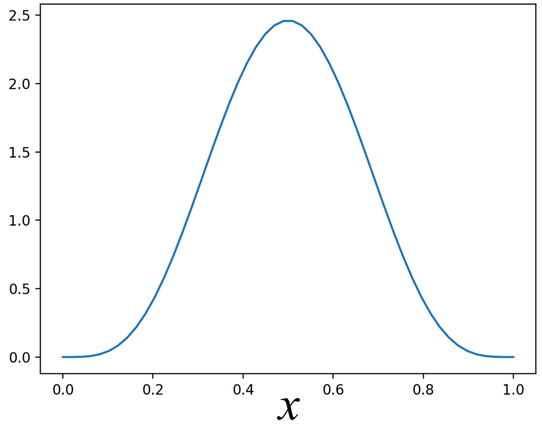
$\alpha = \beta = 0$



$\alpha = \beta = 2$



$\alpha = \beta = 5$



Beta distribution:

$$\mathcal{B}(x|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

with $$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

$$\langle x \rangle = \frac{\alpha}{\alpha+\beta}$$

$$\mathrm{Var}[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Is the coin fair? - including a beta prior

$$\mathcal{P}(p|N_h, N) \propto \mathcal{P}(N_h|p, N)\mathcal{P}(p) \propto \binom{N}{N_h} p^{N_h}(1-p)^{N_t}\mathcal{P}(p)$$
$$\propto \mathcal{B}(p|\alpha = N_h + 1, \beta = N_t + 1)\mathcal{P}(p)$$
$$\propto \mathcal{B}(p|\alpha = N_h + 1, \beta = N_t + 1)\mathcal{B}(p|\alpha = a, \beta = a)$$
$$\propto \mathcal{B}(p|\alpha = N_h + a, \beta = N_t + a)$$

Probability to get heads: p
Tails: $N_t = N - N_h$

This gives a pseudo-count of a.

The best estimate (the average) of p:

$$\langle p \rangle = \frac{N_h + a}{(N_h + a) + (N_t + a)} = \frac{N_h + a}{N + 2a}$$

Pseudo-count of a heads and a tails

Beta distribution:

$$\mathcal{B}(x|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

with $$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

$$\langle x \rangle = \frac{\alpha}{\alpha + \beta}$$

$$\mathrm{Var}[x] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

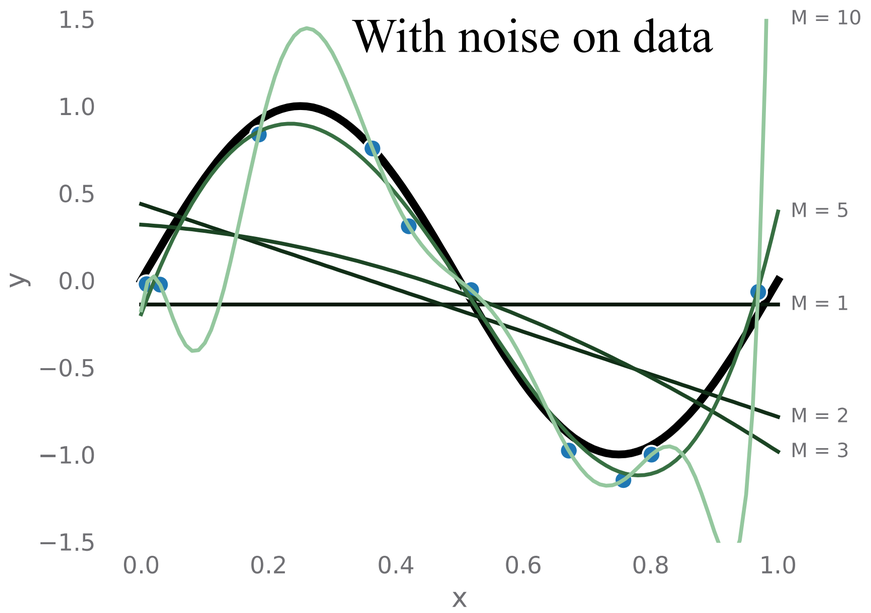
Machine learning

Method of least squares

How do we determine the best fit, i.e. the parameters w_i?

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1}$$

Function values: $f(x_n)$ (not directly known)
Measured values: $t_n = f(x_n) + \text{noise}$



Method of least squares:

Minimize the *cost function C*:

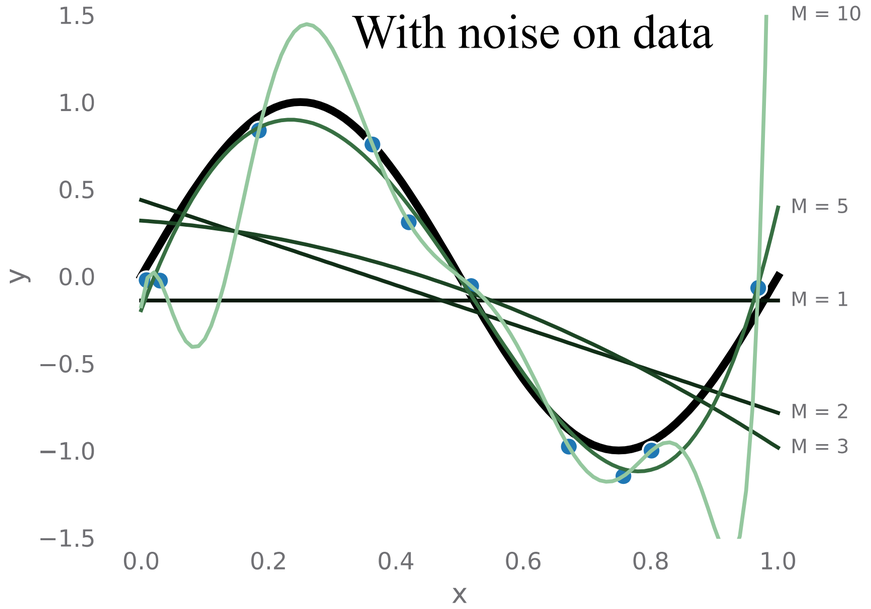
$$C(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - y(x_n, \boldsymbol{w}))^2$$

From Bayes to least squares

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1}$$

Function values: $f(x_n)$ (not directly known)
Measured values: $t_n = f(x_n) + \text{noise}$

Assume, data is Gaussianly distributed with noise σ: $\mathcal{N}(t_n | f(x_n), \sigma^2)$



The probability distribution for the parameters becomes

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) = \prod_{n=1}^{N} \left(\frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(t_n - y(x_n, \boldsymbol{w}))^2 / 2\sigma^2} \right) \mathcal{P}(\boldsymbol{w})$$

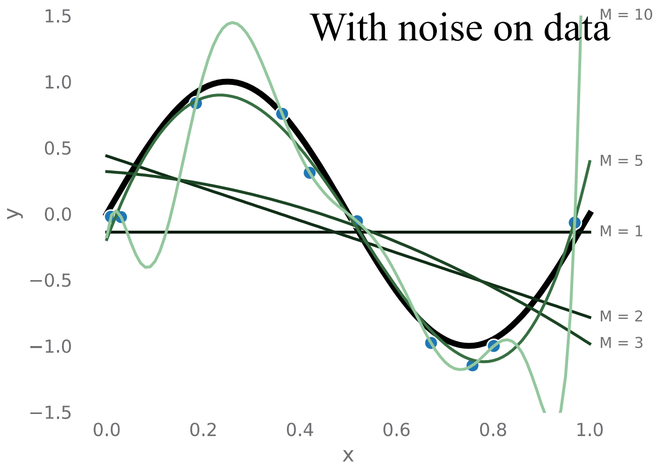
$$= \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-\sum_n (t_n - y(x_n, \boldsymbol{w}))^2 / 2\sigma^2} \mathcal{P}(\boldsymbol{w})$$

Maximize the likelihood -> Minimize the cost function

$$C(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - y(x_n, \boldsymbol{w}))^2$$

Linear basis function models

Linear model with basis functions: $y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$ $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_M)$



Probability distribution of parameters:

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) = \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) = \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-\sum_n (t_n - y(x_n, \boldsymbol{w}))^2 / 2\sigma^2} \mathcal{P}(\boldsymbol{w})$$

$$= \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-C(\boldsymbol{w})/\sigma^2} \mathcal{P}(\boldsymbol{w})$$

with cost function $C(\boldsymbol{w}) = \frac{1}{2} \sum_n (t_n - \sum_i w_i \phi_i(\boldsymbol{x}_n))^2 = \frac{1}{2} (\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})^2$

$\boldsymbol{\Phi}_{nj} = \phi_j(x_n)$ is the so-called *design matrix*. $N \times M$ matrix

N data points
M basis functions

First step: Neglect prior, and maximize the likelihood → Least squares method

Least squares method - exercise

Linear model with basis functions: $y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$ $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_M)$

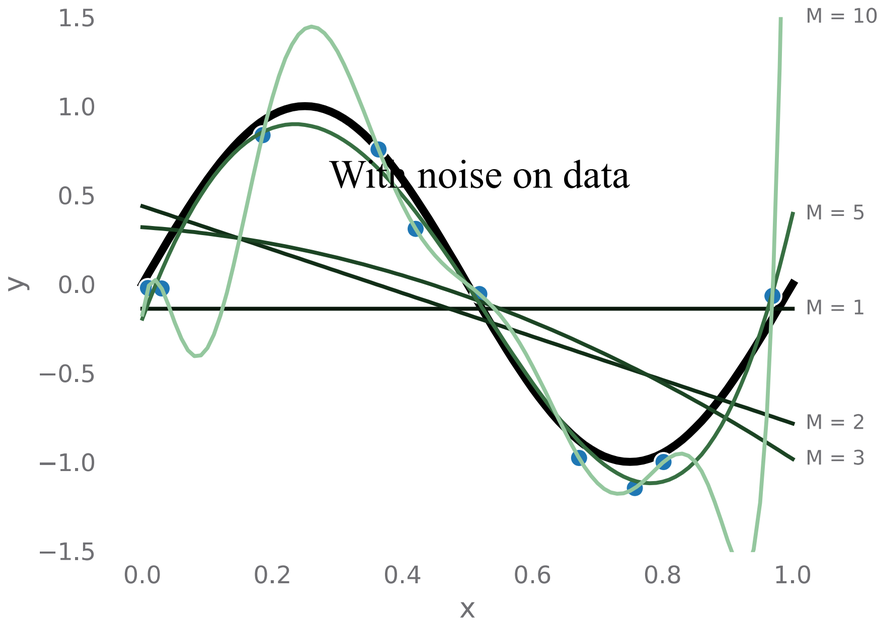
Minimize cost function: $C(\boldsymbol{w}) = \frac{1}{2} \sum_n (t_n - \sum_i w_i \phi_i(\boldsymbol{x}_n))^2 = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})^2$ $\boldsymbol{\Phi}_{nj} = \phi_j(x_n)$

$$\frac{\partial C}{\partial \boldsymbol{w}} = \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{w} - \boldsymbol{\Phi}^T \boldsymbol{t} = 0$$

Solution: $\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$

Moore-Penrose inverse of $\boldsymbol{\Phi}$.

Do the exercise in Chapter 13 in the notes.



Least squares method - lessons learned

- Very simple to solve by matrix inversion
- Works well for simple models (small M)
- Overfitting for large M and parameters ($\mathbf{w}$) become huge

Regularized least squares - including the prior

We would expect in advance that the coefficients **w** should not be huge.
(A Taylor expansion of the sine function will give coefficients of the order one.)

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) \propto e^{-\sum_n (t_n - y(x_n,\boldsymbol{w}))^2/2\sigma^2}\mathcal{P}(\boldsymbol{w})$$
$$\propto e^{-C(\boldsymbol{w})/\sigma^2}\mathcal{P}(\boldsymbol{w}) \propto e^{-C_{\mathrm{reg}}(\boldsymbol{w})/\sigma^2}$$

Take $\mathcal{P}(\boldsymbol{w}) \propto e^{-\frac{1}{2}\lambda \boldsymbol{w}^T\boldsymbol{w}} = e^{-\frac{1}{2}\lambda\sum_i |w_i|^2}$ ⟶ $C_{\mathrm{reg}}(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{t}-\boldsymbol{\Phi w})^2 + \frac{1}{2}\tilde{\lambda}\boldsymbol{w}^T\boldsymbol{w}$

$\tilde{\lambda} = \sigma^2\lambda$

Maximizing the *posterior distribution* corresponds to minimizing the regularized cost function. This gives

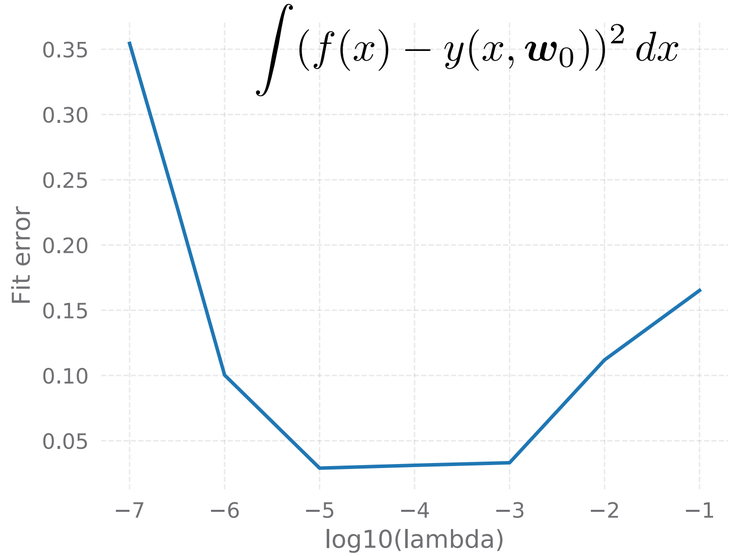
$$\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}$$

Now do the exercise on the regularized least squares method!

Regularized least squares - lessons learned

- The regularization makes it possible to control the size of the parameters, so they do not go crazy.
- This allows for more parameters, because the regularization is controlling the complexity of the model.

- Figure shows calculated error versus $\log_{10}(\tilde{\lambda})$.



Multi-dimensional Gaussian distribution

Consider two variables: y_1, y_2

Assume the two variables are Gaussianly distributed with zero mean and some correlation between them:

$$\langle y_1^2 \rangle = \sigma^2,\ \langle y_2^2 \rangle = \sigma^2,\ \langle y_1 y_2 \rangle = \tau^2\ \ or\ \langle \mathbf{y}\mathbf{y}^\mathbf{T} \rangle = \begin{pmatrix} \sigma^2 & \tau^2 \\ \tau^2 & \sigma^2 \end{pmatrix} \equiv \mathbf{K}$$

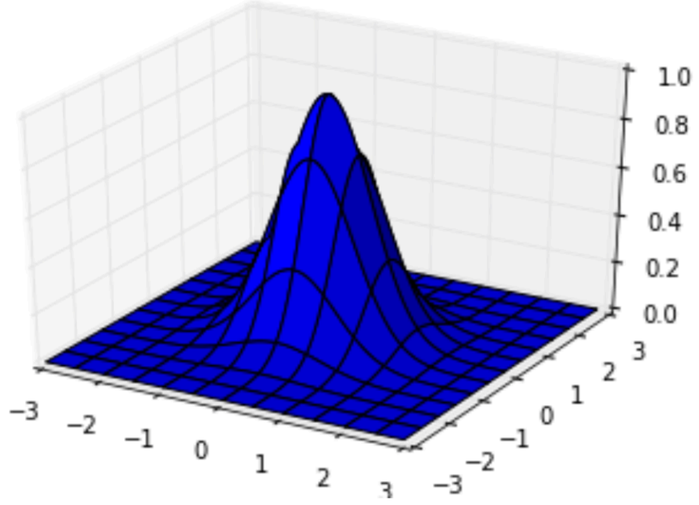
This corresponds to the probability distribution:

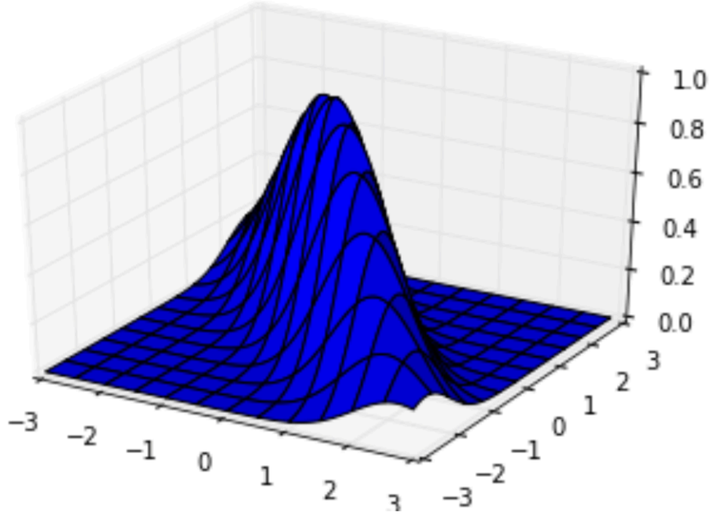
$$P_0(y_1, y_2) = \frac{1}{\sqrt{2\pi det(\mathbf{K})}} \exp\left(-\frac{1}{2}\mathbf{y}^\mathbf{T}\mathbf{K}^{-1}\mathbf{y}\right)$$

No correlation $\tau = 0$

Strong correlation $\tau \sim \sigma$

$P(y_1, y_2)$





Multi-dimensional Gaussian distribution

Now we get the information that y_1 actually has the value y_1^0 .
What is then the probability distribution for y_2?

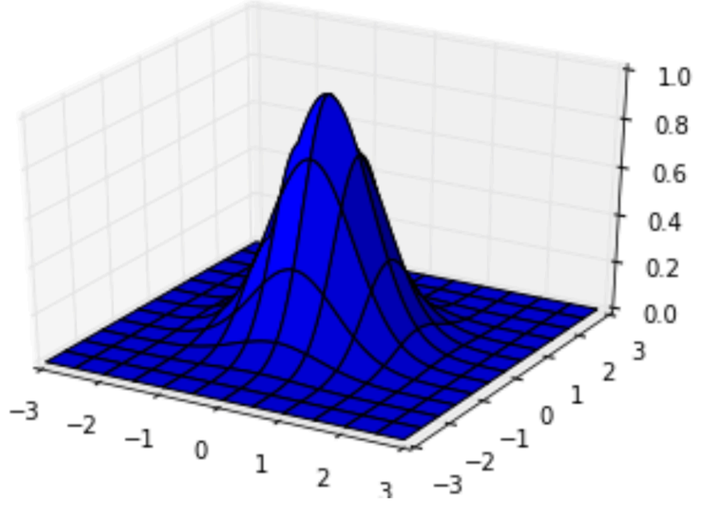
$$P(y_2) \propto P_0(y_1^0, y_2) \propto \exp\left[-\left(y_2 - \left(\frac{\tau}{\sigma}\right)^2 y_1^0 \right)^2 \Big/ 2(\sigma^2(1-(\tau/\sigma)^4)) \right]$$

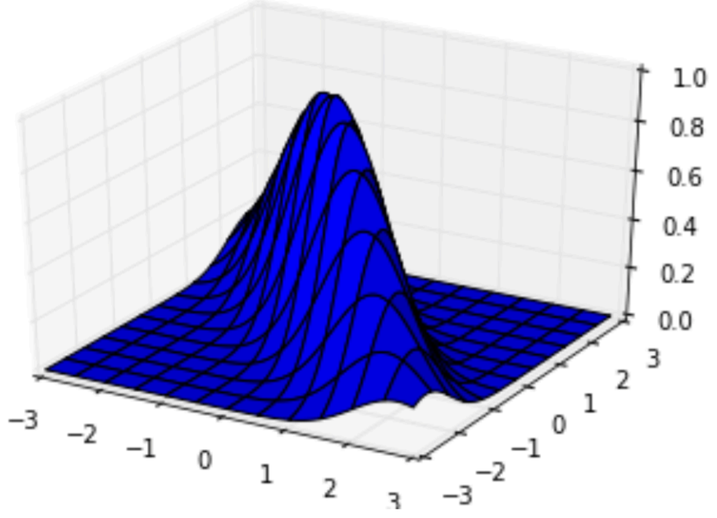
A new Gaussian! Mean: $\langle y_2 \rangle = \left(\frac{\tau}{\sigma}\right)^2 y_1^0$ Variance: $\langle (y_2 - \langle y_2 \rangle)^2 \rangle = \sigma^2(1-(\tau/\sigma)^4)$

No correlation $\tau = 0$

Strong correlation $\tau \sim \sigma$

$P(y_1, y_2)$





See appendix in notes

Multi-dimensional Gaussian distribution

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{\det(\Sigma)^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$$

$$\langle \boldsymbol{x} \rangle = \boldsymbol{\mu} \quad \text{and} \quad \text{Cov}[\boldsymbol{x}, \boldsymbol{x}^T] = \boldsymbol{\Sigma}$$

Convolution:

$$\mathcal{P}(\boldsymbol{t}) = \int \mathcal{N}(\boldsymbol{t}|\boldsymbol{y},\boldsymbol{\Sigma})\mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p,\boldsymbol{K})\,d\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p,\boldsymbol{C}) \qquad \text{where} \qquad \boldsymbol{C} = \boldsymbol{K} + \boldsymbol{\Sigma}$$

Product:

$$\mathcal{N}(\boldsymbol{y}|\tilde{\boldsymbol{y}},\boldsymbol{K}) \propto \mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_1,\boldsymbol{K}_1)\mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_2,\boldsymbol{K}_2) \quad \text{where} \quad \boldsymbol{K}^{-1} = \boldsymbol{K}_1^{-1} + \boldsymbol{K}_2^{-1} \text{ and } \tilde{\boldsymbol{y}} = \boldsymbol{K}\left(\boldsymbol{K}_1^{-1}\boldsymbol{y}_1 + \boldsymbol{K}_2^{-1}\boldsymbol{y}_2.\right)$$

Fully Bayesian: the posterior distribution

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) = \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) \propto e^{-\sum_n (t_n - y(x_n,\boldsymbol{w}))^2/2\sigma^2} e^{-\frac{1}{2}\lambda \boldsymbol{w}^T\boldsymbol{w}} = e^{-C_{reg}(\boldsymbol{w})/\sigma^2}$$

$$C_{\text{reg}}(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{\Phi w})^2 + \frac{1}{2}\tilde{\lambda}\boldsymbol{w}^T\boldsymbol{w} \qquad \tilde{\lambda} = \sigma^2\lambda$$

Maximize the posterior: $\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}$

Rewrite:
$$\begin{aligned} C_{\text{reg}}(\boldsymbol{w}) &= \frac{1}{2}(\boldsymbol{t} - \boldsymbol{\Phi w})^2 + \frac{1}{2}\tilde{\lambda}\boldsymbol{w}^T\boldsymbol{w} = \frac{1}{2}(\boldsymbol{t}^T - \boldsymbol{w}^T\boldsymbol{\Phi}^T)(\boldsymbol{t} - \boldsymbol{\Phi w}) + \frac{1}{2}\tilde{\lambda}\boldsymbol{w}^T\boldsymbol{w} \\ &= \frac{1}{2}\boldsymbol{w}^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda})\boldsymbol{w} + \text{ linear in } \boldsymbol{w} + \text{constant} \\ &= \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^T\boldsymbol{\Omega}^{-1}(\boldsymbol{w} - \boldsymbol{w}_0) + \text{ constant} \qquad \text{with} \qquad \boldsymbol{\Omega} = \sigma^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1} \end{aligned}$$

So, the posterior can be written

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^T\boldsymbol{\Omega}^{-1}(\boldsymbol{w} - \boldsymbol{w}_0)\right) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Omega})$$

Fully Bayesian: the posterior probability distribution

Posterior:

with $\boldsymbol{\Omega} = \sigma^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}$

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_0)^T\boldsymbol{\Omega}^{-1}(\boldsymbol{w}-\boldsymbol{w}_0)\right) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Omega})$$

$$\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t} = \frac{1}{\sigma^2}\boldsymbol{\Omega}\boldsymbol{\Phi}^T\boldsymbol{t}$$

So, we get for the average and the covariance: $\langle \boldsymbol{w} \rangle = \boldsymbol{w}_0$ and $\mathrm{Cov}[\boldsymbol{w}^T, \boldsymbol{w}] = \boldsymbol{\Omega}$

The predicted functions are given by: $y(x, \boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{\phi}(x)$

So, the best fit function is: $y_0(x, \boldsymbol{w}) = \boldsymbol{w}_0^T\boldsymbol{\phi}(x) = \sum_{i=1}^{M} w_{0,i}\phi_i(x)$

and the variance (i.e. the (square of) uncertainty of the prediction becomes

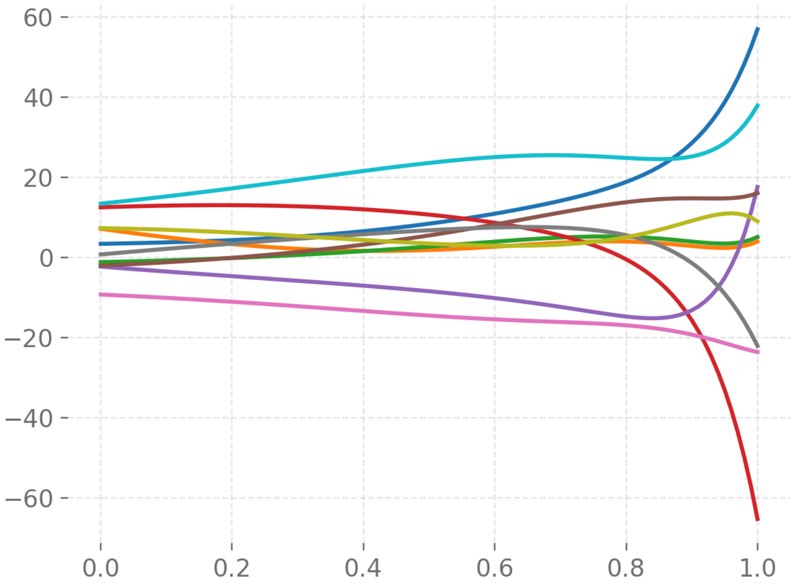
$$\mathrm{Var}[y(x,\boldsymbol{w})] = \mathrm{Cov}[y(x,\boldsymbol{w}), y(x,\boldsymbol{w})] = \mathrm{Cov}[\boldsymbol{\phi}^T(x)\boldsymbol{w}, \boldsymbol{w}^T\boldsymbol{\phi}(x)] = \boldsymbol{\phi}^T(x)\mathrm{Cov}[\boldsymbol{w}, \boldsymbol{w}^T]\boldsymbol{\phi}(x) = \boldsymbol{\phi}^T(x)\boldsymbol{\Omega}\boldsymbol{\phi}(x)$$

Now do the exercise about Bayesian analysis of a linear basis function model!

Prior distribution

$$\mathcal{P}(\boldsymbol{w}) \propto e^{-\frac{1}{2}\lambda \boldsymbol{w}^T \boldsymbol{w}} = e^{-\frac{1}{2}\lambda \sum_i |w_i|^2}$$
$$\propto \mathcal{N}(\boldsymbol{w}|0, 1/\lambda * \boldsymbol{I}_M)$$

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$$




```
ngrid = 101
xg = np.linspace(0,1,num=ngrid)

M = 20
lamb = 10**(-2)  # Do not use lambda here! That is a reserved word in Python.
Nens = 10

def basisf(j,x):
    return x**j

def fitf(x):
    return sum([w[j]*basisf(j,x) for j in range(M)])

for i in range(Nens):
    w=np.random.multivariate_normal(np.zeros(M),np.identity(M)/lamb)
    plt.plot(xg, fitf(xg))
```

Posterior distribution

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_0)^T\boldsymbol{\Omega}^{-1}(\boldsymbol{w}-\boldsymbol{w}_0)\right)$$

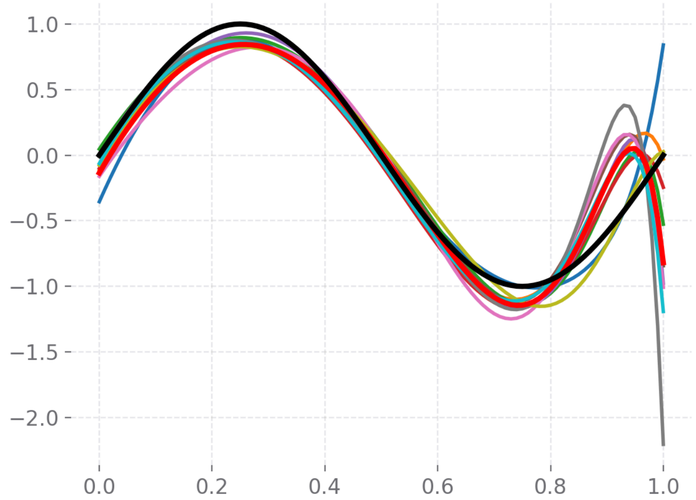
$$= \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Omega})$$

$$\boldsymbol{\Phi}_{nj} = \phi_j(x_n)$$

$$\boldsymbol{w}_0 = \frac{1}{\sigma^2}\boldsymbol{\Omega}\boldsymbol{\Phi}^T\boldsymbol{t}$$

$$\boldsymbol{\Omega} = \sigma^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}$$

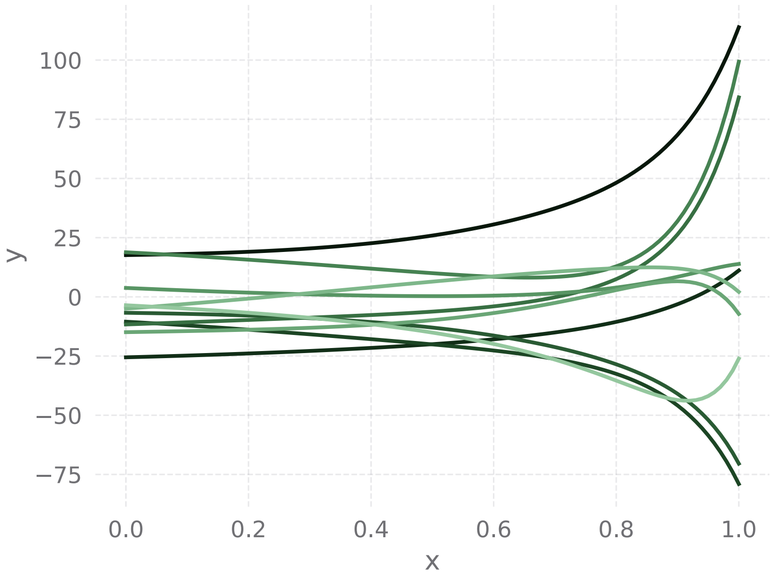
$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} w_i\phi_i(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})$$



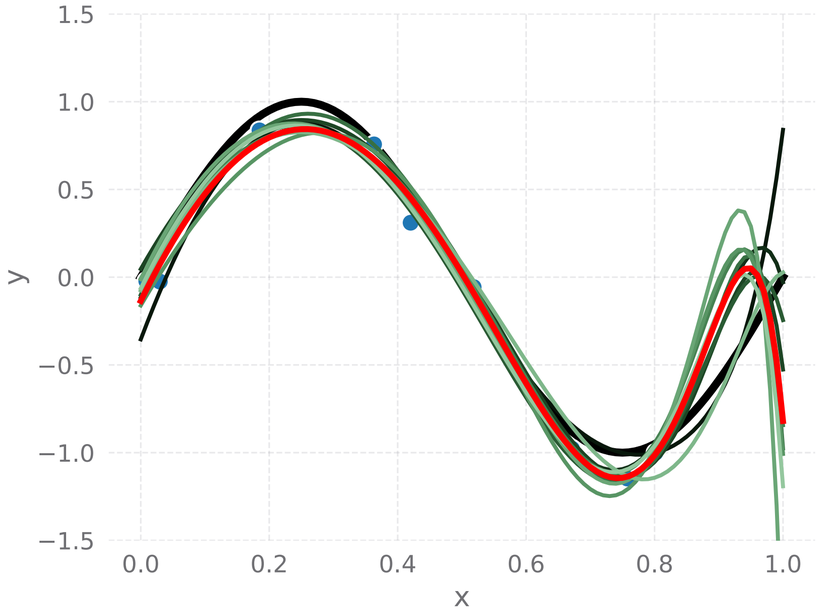

```
initwithnoise()
M = 2*N
lambtilde = 10**(-4)
Nens = 10
sigma = 0.1

ngrid = 101
xg = np.linspace(0,1,num=ngrid)

def basisf(j,x):
    return x**j

def fitf(x):
    return sum([w[j]*basisf(j,x) for j in range(M)])

phi = np.zeros((N,M))
for n, x in enumerate(xp):
    for j in range(M):
        phi[n,j] = basisf(j,x)
phit = phi.transpose()

omega = sigma**2*np.linalg.inv(phit @ phi + lambtilde*np.identity(M))

w0 = np.dot(omega/sigma**2, np.dot(phi.transpose(),tp))

for i in range(Nens):
    w=np.random.multivariate_normal(w0,omega)
    plt.plot(xg, fitf(xg))
w=w0
plt.plot(xg, fitf(xg), color='r', lw=3)
plt.plot(xg,np.sin(2*np.pi*xg), color='k', lw=3)
```

Exercise about Bayesian analysis of a linear basis function model - lessons learned

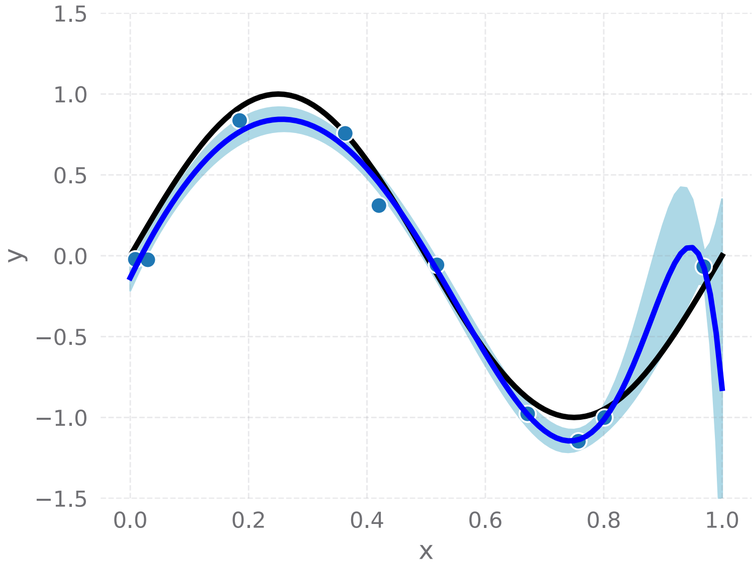
Ensemble of functions drawn from the prior distribution



Ensemble of functions drawn from the posterior distribution



Best fit function with error bar (one standard deviation)



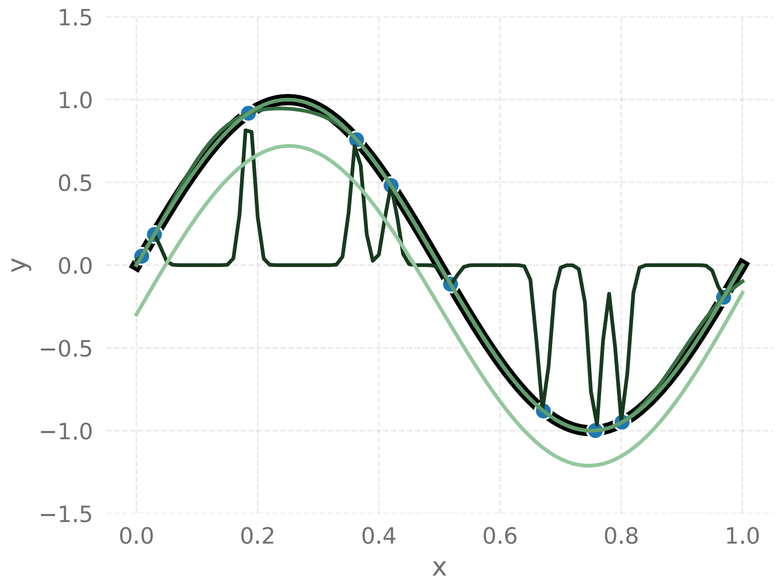
And now for something completely different: Kernel regression

We now choose the basis functions as a Gaussian function *at each data point.*

$$\boldsymbol{g}(x)^T = (g(x, x_1), g(x, x_2), \ldots, g(x, x_N))$$

$$g(x, x') = \exp(-(x - x')^2 / 2\ell^2)$$

$$y(x, \boldsymbol{a}) = \sum_{n=1}^{N} g(x, x_n) a_n = \boldsymbol{g}(x)^T \boldsymbol{a}$$



Same number of data points and parameters!

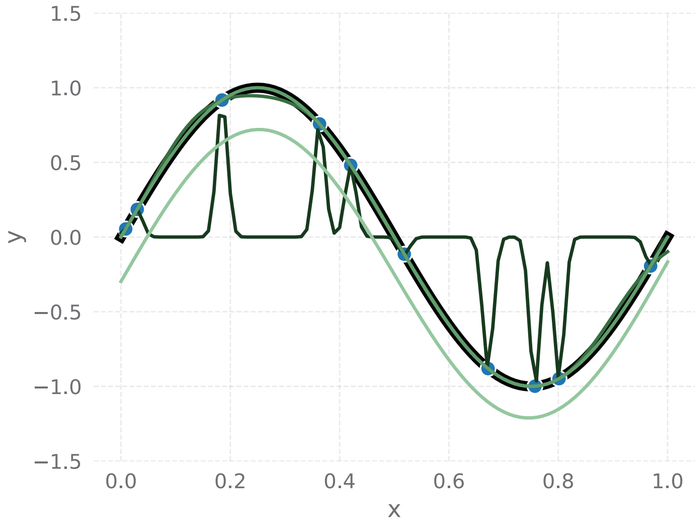
$$\boldsymbol{G}_{nm} = g(x_n, x_m) \qquad \boldsymbol{y} = \boldsymbol{G}\boldsymbol{a}_0 \implies \boldsymbol{a}_0 = \boldsymbol{G}^{-1}\boldsymbol{y}$$

Prediction: $y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{a}_0 = \boldsymbol{g}(x)^T \boldsymbol{G}^{-1} \boldsymbol{y}$

Exercise on kernel regression - lessons learned

$$g(x, x') = \exp(-(x - x')^2 / 2\ell^2)$$

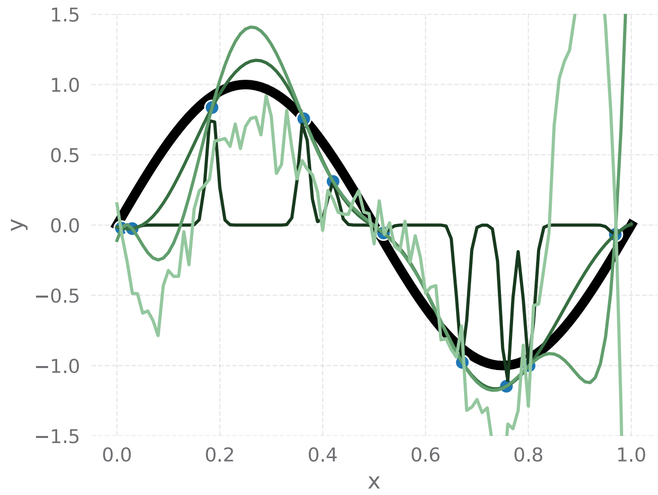
Without noise



$\ell = 0.01, 0.1, 0.5$ and 1.0

Length scale very important for quality of fit!

With noise



All noise is fitted. Severe problems for both short and long scales

Kernel ridge regression

What if we have noise on the data $\mathbf{t} = \mathbf{y} +$ noise ?

The solution from before reproduces all noise! $y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{a}_0 = \boldsymbol{g}(x)^T \boldsymbol{G}^{-1} \boldsymbol{t}$

Invent a regularized cost function: $C_{\text{reg}} = (\boldsymbol{t} - \boldsymbol{G}\boldsymbol{a})^2 + \lambda' \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a}$

New solution: $\boldsymbol{a}_0 = (\boldsymbol{G} + \lambda' \boldsymbol{I}_M)^{-1} \boldsymbol{t}$

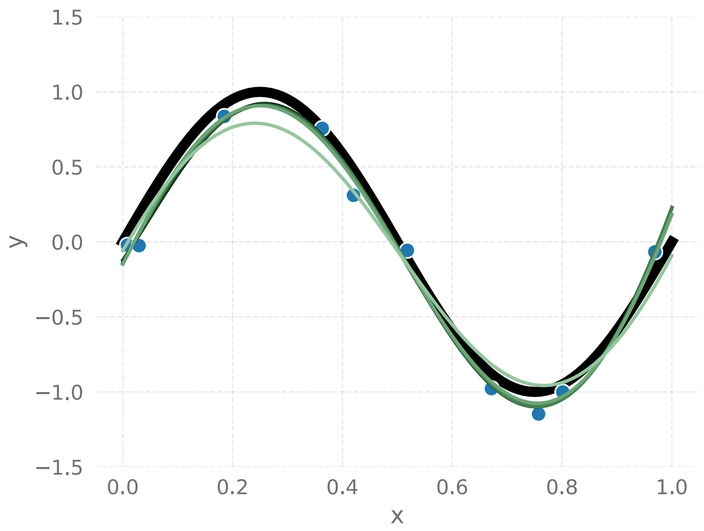
New prediction: $y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{a}_0 = \boldsymbol{g}(x)^T (\boldsymbol{G} + \lambda' \boldsymbol{I}_N)^{-1} \boldsymbol{t}$

Now do the exercises on kernel regression and kernel ridge regression!

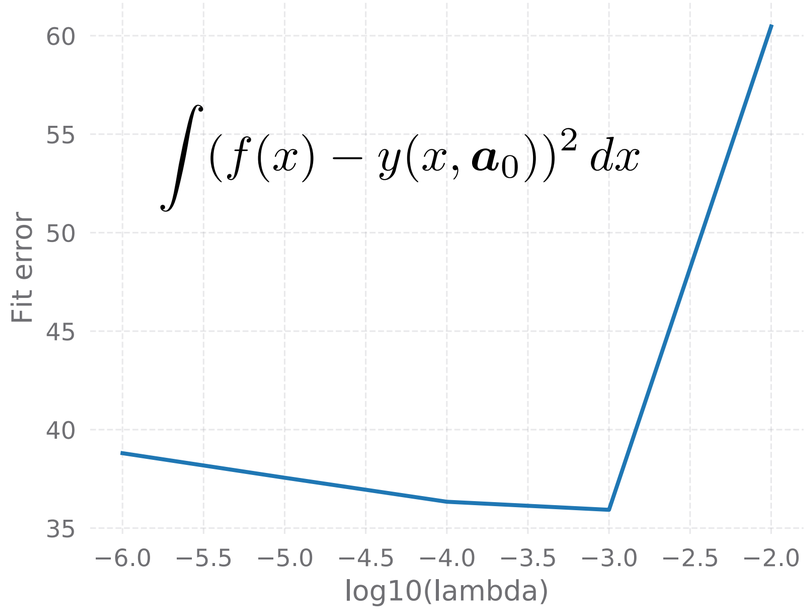
Exercise on kernel ridge regression - lessons learned

$\ell = 0.5$

$\lambda' = 10^{-6} - 10^{-2}$



Error as a function of regularization



Really nice and stable fits!

Today

- 13:00 Gaussian processes - theory
- 13:30 Exercises on GP
- 15:00 Hyperparameters - theory
- 15:30 Exercises on hyperparameters
-

Gaussian processes - the real thing!

- Includes the linear basis functions models, but are more general
- Can reproduce the results of kernel ridge regression
- Includes full probability distributions, so can predict uncertainties

Gaussian processes - derivation from linear basis function models

Work with parameters $\boldsymbol{w}$

Basis functions: $\phi_i(\boldsymbol{x})$

$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$$

Prior distribution: $\mathcal{P}(\boldsymbol{w}) \propto \exp(-\lambda \boldsymbol{w}^T \boldsymbol{w}/2)$

Average: $\langle \boldsymbol{w} \rangle = 0$

Covariance: $\mathrm{Cov}[\boldsymbol{w}, \boldsymbol{w}^T] = \frac{1}{\lambda} \boldsymbol{I}_M$

Prior distribution:

$$\mathcal{P}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \frac{1}{\lambda}\boldsymbol{I}_M) \propto \exp(-\lambda \boldsymbol{w}^T \boldsymbol{w}/2)$$

Work directly with function $y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$

Kernel: $k(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{\lambda} \sum_{i=1}^{M} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}')$

Prior distribution: $\mathcal{P}([y(\boldsymbol{x})])$?

Average: $\langle y(\boldsymbol{x}) \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w} \rangle = 0$

Covariance:

$$\mathrm{Cov}[y(\boldsymbol{x}), y(\boldsymbol{x}')] = \langle y(\boldsymbol{x}) y(\boldsymbol{x}') \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w}\boldsymbol{w}^T \rangle \boldsymbol{\phi}(\boldsymbol{x}')$$

$$= \frac{1}{\lambda} \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{I}_M \boldsymbol{\phi}(\boldsymbol{x}') = \frac{1}{\lambda} \sum_{i=1}^{M} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}')$$

Prior distribution:

$$\mathcal{P}([y(\boldsymbol{x})]) = \mathcal{N}([y(\boldsymbol{x})]|0, k(\boldsymbol{x}, \boldsymbol{x}'))$$

Gaussian process - the prior distribution

Prior distribution:

$$\mathcal{P}([y(\boldsymbol{x})]) = \mathcal{N}([y(\boldsymbol{x})]|0, k(\boldsymbol{x}, \boldsymbol{x}'))$$

$$\propto \exp\left(-\frac{1}{2}\int y(\boldsymbol{x})k^{-1}(\boldsymbol{x}, \boldsymbol{x}')y(\boldsymbol{x}')\, d\boldsymbol{x}\, d\boldsymbol{x}'\right)$$

Kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{\lambda}\sum_{i=1}^{M}\phi_i(\boldsymbol{x})\phi_i(\boldsymbol{x}')$$

but now we will forget about the basis functions!

"Inverse" function defined like matrix inverse

$$\int k^{-1}(\boldsymbol{x}, \boldsymbol{x}'')k(\boldsymbol{x}'', \boldsymbol{x}')\, d\boldsymbol{x}'' = \delta(\boldsymbol{x} - \boldsymbol{x}')$$

We shall now look at the Gaussian ("squared-exponential") kernel: $k(x, x') = \exp(-(x - x')^2/2\ell^2)$

Interpretation of kernel as covariance matrix!
$y(\mathbf{x})$ and $y(\mathbf{x}')$ are *correlated* if $\mathbf{x}$ and $\mathbf{x}'$ are close and *uncorrelated* or *independent* if they are far away from each other.

$$\mathrm{Cov}[y(\boldsymbol{x}), y(\boldsymbol{x}')] = \langle y(\boldsymbol{x})y(\boldsymbol{x}')\rangle = k(\boldsymbol{x}, \boldsymbol{x}')$$

(Do the exercise on the prior distribution with Gaussian kernel function.)

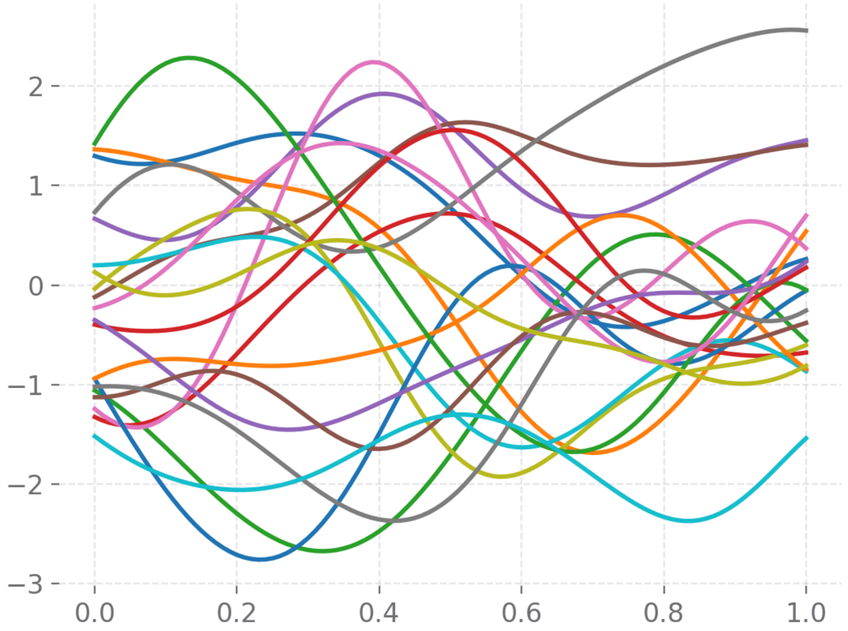
Prior distribution:

$$\mathcal{P}([y(\boldsymbol{x})]) = \mathcal{N}([y(\boldsymbol{x})]|0, k(\boldsymbol{x}, \boldsymbol{x}'))$$
$$\propto \exp\left(-\frac{1}{2}\int y(\boldsymbol{x})k^{-1}(\boldsymbol{x}, \boldsymbol{x}')y(\boldsymbol{x}')\, d\boldsymbol{x}\, d\boldsymbol{x}'\right)$$

Kernel: $k(x, x') = \exp(-(x - x')^2/2\ell^2)$

$\mathrm{Cov}[y(\boldsymbol{x}), y(\boldsymbol{x}')] = \langle y(\boldsymbol{x})y(\boldsymbol{x}')\rangle = k(\boldsymbol{x}, \boldsymbol{x}')$

```
xg = np.linspace(0,1,101)

ll = 0.2
def kernel(x,xx):
    return np.exp(-(x-xx)**2/(2*ll**2))

Nens = 20

k = np.array([[kernel(x,xx) for x in xg] for xx in xg ])

for i in range(Nens):
    y = np.random.multivariate_normal(np.zeros(ngrid),k)
    plt.plot(xg,y)
```

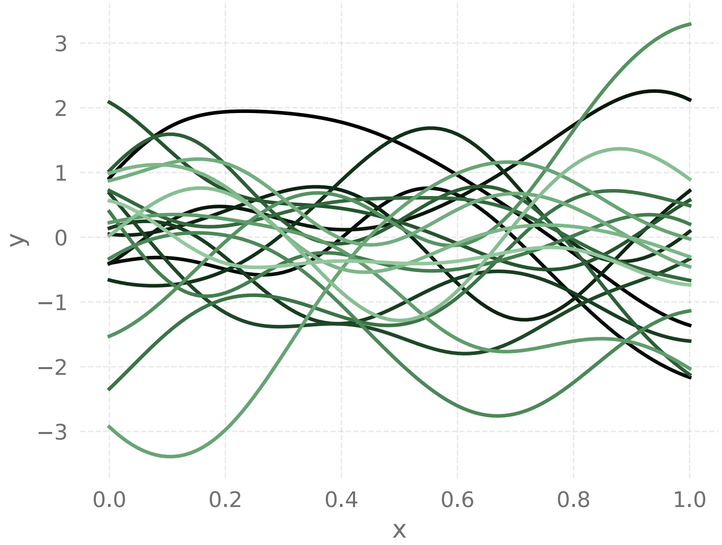


Do the exercise on the prior distribution with Gaussian kernel function.

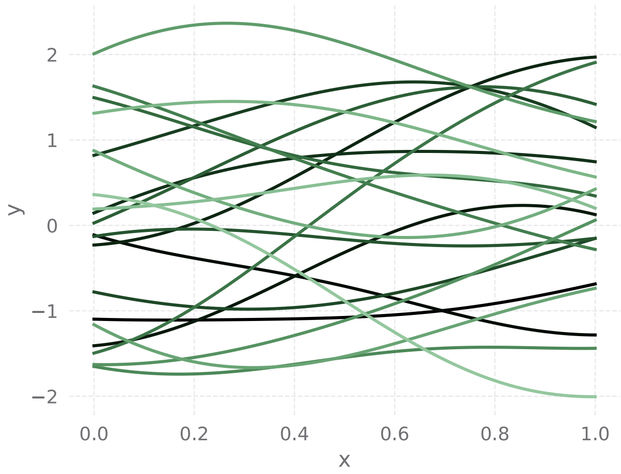
Gaussian process - the prior distribution - lessons learned

$$k(x, x') = \exp(-(x - x')^2 / 2\ell^2)$$

Ensemble of functions with $\ell = 0.2$



Ensemble of functions with $\ell = 0.6$



The length scale determines our expectations about the functions.
Much nicer functions than the ones we obtained with the polynomial basis.

Gaussian process - prediction

From the linear basis function models, we have the prediction

$$y_0(\boldsymbol{x}) := \langle y(\boldsymbol{x}) \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w} \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}_0 = \boldsymbol{\phi}(\boldsymbol{x})^T (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \tilde{\lambda} \boldsymbol{I}_M)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}$$

basis functions — design matrix — data

This can be rewritten (see the notes)

$$y_0(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{t}$$

$$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

$$k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n) \qquad \boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$$

We do not need the basis functions. The prediction is expressed using the kernel function. Prediction looks exactly like kernel ridge regression, if the function g is identified with the kernel.

Gaussian process - variance

From the linear basis function models, we have

$$\sigma^2(\boldsymbol{x}) := \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle = \boldsymbol{\phi}^T(x)\sigma^2 \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M \right)^{-1} \boldsymbol{\phi}(x)$$

This can be rewritten (see the notes)

$$\sigma^2(\boldsymbol{x}) = \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{k}(\boldsymbol{x})$$

Also the variance is expressed exclusively using the kernel!

$$k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n)$$

$$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

$$\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$$

Gaussian process

Gaussian process. Given the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ we can define the vector function $\boldsymbol{k}(\boldsymbol{x})$ by $k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n)$ and the kernel matrix $\boldsymbol{K}$ by $K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_n)$. We furthermore define the matrix $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$. The prior average function is denoted $y_p(\boldsymbol{x})$ with the corresponding vector $(\boldsymbol{y}_p)_n = y_p(\boldsymbol{x}_n)$. With these definitions we get the prediction for the average function $y_0(\boldsymbol{x})$ and the variance $\sigma^2(\boldsymbol{x})$ as

$$y_0(\boldsymbol{x}) = y_p(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1}(\boldsymbol{t} - \boldsymbol{y}_p) \tag{17.16}$$

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{k}(\boldsymbol{x}) \tag{17.17}$$

What happens, if we try to make a prediction far away from the data points?

Gaussian process

Kernel: $k(x, x') = k_0 \exp(-(x - x')^2/2\ell^2)$

$k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n)$

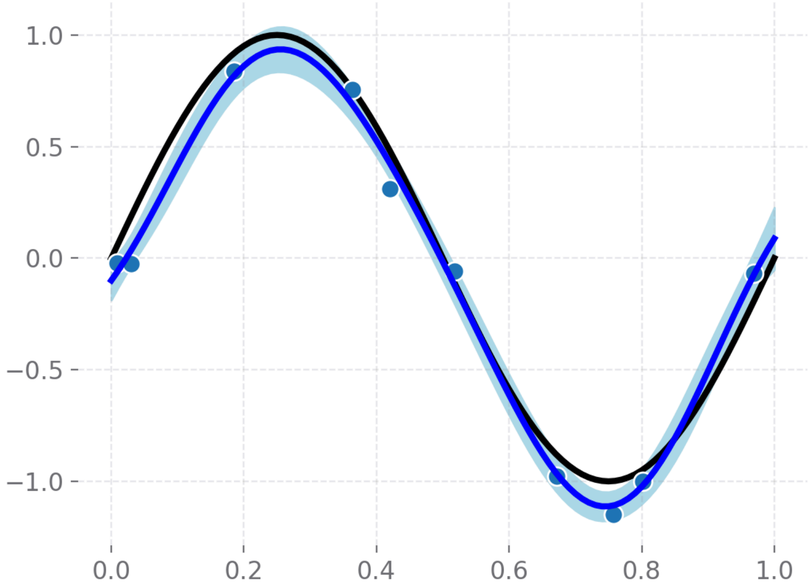
$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$

$\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$

Prediction: $y_0(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{t}$

Variance: $\sigma^2(\boldsymbol{x}) = \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle$

$= k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{k}(\boldsymbol{x})$




```
initwithnoise()
# The x-values of the data are in the array "xp"
# and the t-values are in the array tp

ngrid = 101
xg = np.linspace(0,1,ngrid)
sigma = 0.1

ll = 0.2
k0 = 1

def kernel(x,xx):
    return k0*np.exp(-(x-xx)**2/(2*ll**2))

def kvec(x):
    return np.array([kernel(x,xx) for xx in xp])

K = np.array([[kernel(x,xx) for x in xp] for xx in xp])
C = K + sigma**2*np.identity(N)
Cinv = np.linalg.inv(C)
Cinvt = np.dot(Cinv,tp)

def fitf(x):
    return np.dot(kvec(x),Cinvt)

def varx(x):
    return kernel(x,x)- np.dot(kvec(x),np.dot(Cinv,kvec(x)))


plt.plot(xg,np.sin(2*np.pi*xg), color='k', lw=3) # Plot sine function
plt.plot(xp,tp,'o')  # Plot data points

fitg = np.array([fitf(x) for x in xg]) # Fit function on the grid
width = np.array([np.sqrt(varx(x)) for x in xg]) # Width on the grid
plt.plot(xg, fitg, color='b', lw=3)
plt.fill_between(xg, fitg-width, fitg+width, color='lightblue')
```

Now do the exercise on the Gaussian process for the sine function.

Gaussian process for the sine function - lessons learned

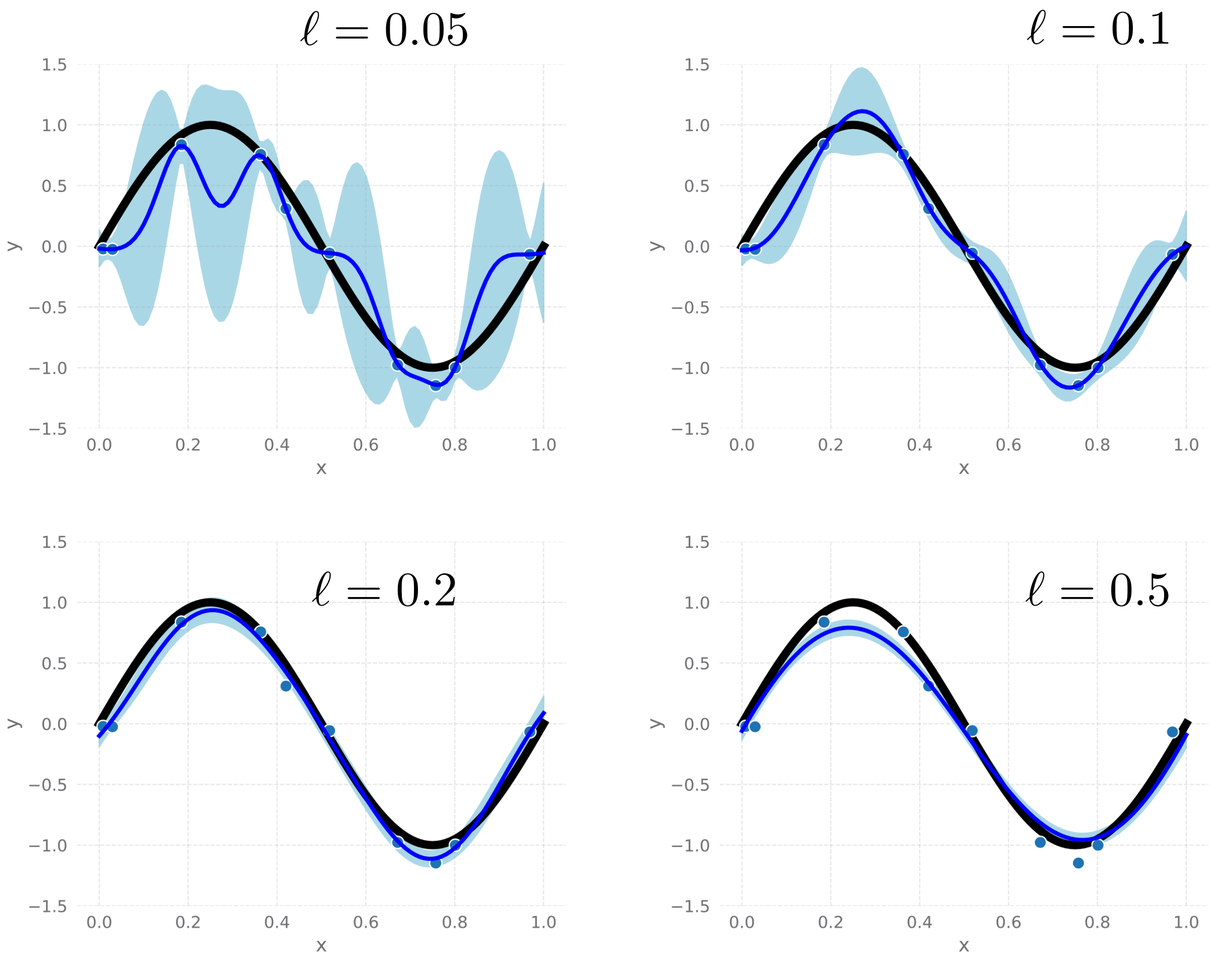
Noise $\sigma = 0.1$

Kernel $k(x, x') = k_0 \exp(-(x - x')^2/2\ell^2)$

$$k_0 = 1 = \langle y(x)^2 \rangle_{\text{prior}}$$

Too small ℓ: poor interpolation between points, large variance

Too large ℓ: model too "stiff", underestimation of errors



Hyperparameters

$$k(x, x') = k_0 \exp(-(x - x')^2/2\ell^2)$$

Examples: noise σ, length scale ℓ, prefactor k_0

$$\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$$

Different strategies to determine the hyperparameters

$$k_0 = 1 = \langle y(x)^2 \rangle_{\text{prior}}$$

1. Prior information
   - Noise might be known
   - The scale ℓ of the problem might be known
   - Prefactor may come from expectations for the variance $k_0 = \langle y(x)^2 \rangle_{\text{prior}}$
2. Cross validation
   - Only use part of the data for “training” and validate on the rest
   - 10-fold cross validation: divide data in 10 sets, train on 9 and test on the last one
   - Pick the values of the hyperparameters that do the best in the cross validation
3. Bayesian analysis (yes!)

Bayesian analysis of hyperparameters for a Gaussian process

Hyperparameters θ

marginalization over **y**

Bayes' theorem:

$$\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta}) = \int \mathcal{P}(\boldsymbol{t}, \boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y}\ \mathcal{P}(\boldsymbol{\theta})$$

$$= \int \mathcal{P}(\boldsymbol{t}|\boldsymbol{y})\mathcal{P}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y}\ \mathcal{P}(\boldsymbol{\theta})$$

$$= \int \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \sigma^2 \boldsymbol{I}_N)\mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p, \boldsymbol{K})\, d\boldsymbol{y}\ \mathcal{P}(\boldsymbol{\theta})$$

$$= \int \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C})\, d\boldsymbol{y}\ \mathcal{P}(\boldsymbol{\theta}) \qquad \boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$$

Too heavy, to work with the full posterior distribution, so we maximize the log-likelihood:

$$\mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\det(\boldsymbol{C})^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{t} - \boldsymbol{y}_p)^T \boldsymbol{C}^{-1}(\boldsymbol{t} - \boldsymbol{y}_p)\right)$$

$$\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = \log \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C})$$

$$= -\frac{1}{2}\log\left(\det(\boldsymbol{C})\right) - \frac{1}{2}(\boldsymbol{t} - \boldsymbol{y}_p)^T \boldsymbol{C}^{-1}(\boldsymbol{t} - \boldsymbol{y}_p) - \frac{N}{2}\log(2\pi)$$

Bayesian analysis of hyperparameters for a Gaussian process

$$\mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\det(\boldsymbol{C})^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{y}_p)\right)$$

$$\begin{aligned}\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) &= \log \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}) \\ &= -\frac{1}{2}\log\left(\det(\boldsymbol{C})\right) - \frac{1}{2}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{y}_p) - \frac{N}{2}\log(2\pi)\end{aligned}$$

Maximize by setting derivative to zero:

$$\frac{\partial}{\partial \theta_i}\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \theta_i}\right) + \frac{1}{2}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \theta_i}\boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{y}_p) = 0$$

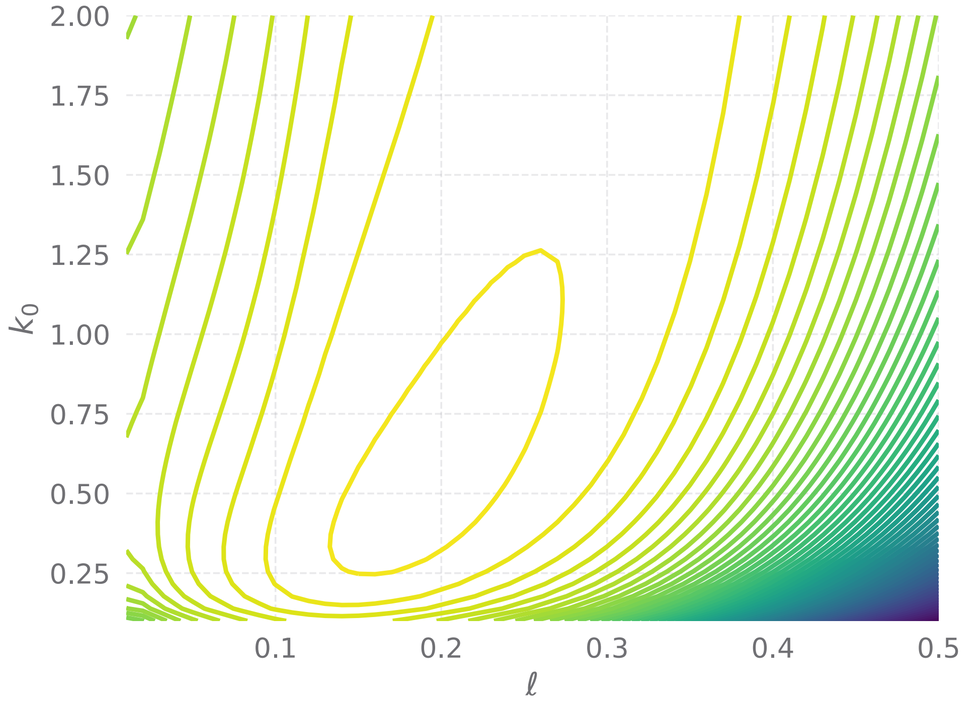
consider fixed

Example: kernel prefactor $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N = k_0 \boldsymbol{K}_0 + \sigma^2 \boldsymbol{I}_N = k_0(\boldsymbol{K}_0 + \frac{\sigma^2}{k_0}\boldsymbol{I}_N) = k_0 \boldsymbol{C}_0$

Maximizing the likelihood gives: $k_0 = \frac{1}{N}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}_0^{-1}(\boldsymbol{t}-\boldsymbol{y}_p)$

Do the exercise on the Bayesian determination of hyperparameters.

Optimizing hyperparameters for sine problem



Too large ℓ requires larger k_0.