Introduction to Artificial Intelligence (ENSIMAG) Intelligent Systems (MOSIG)

Regularization

Original Slides by Clovis Galiez
Lecture: Sergi Pujades

2021-2022

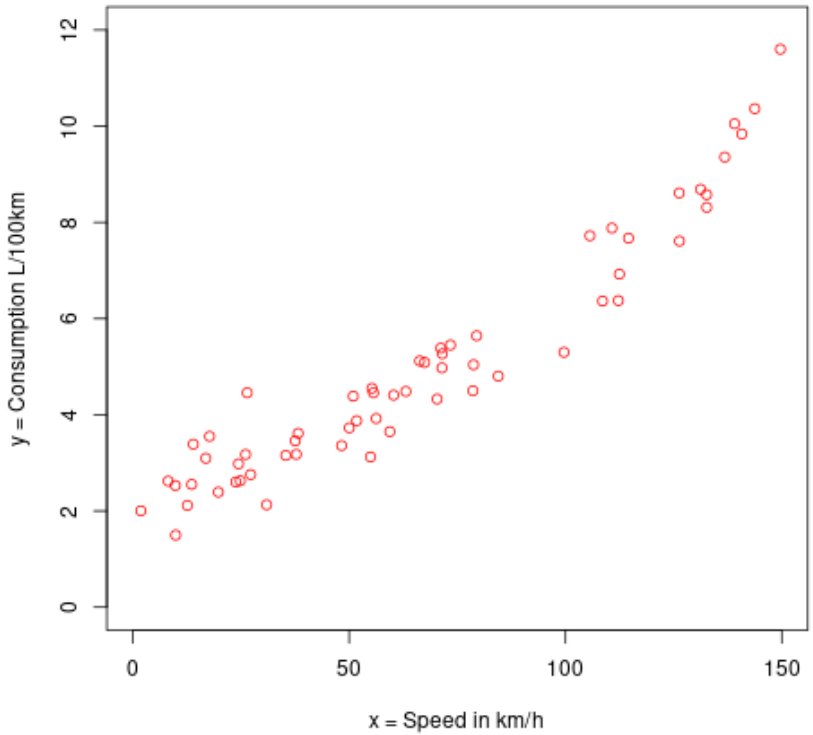
Worked-out example

You want to model the fuel consumption (L/100km) with respect to the speed (km/h).

Worked-out example: the training data

Your neighbor, gives you her home-made measurements.

It consists in $(x_i, y_i), i = 1, ..60$



Worked-out example: choose the model

We want to model the **fuel consumption** y (L/100km) with respect to the **car speed** x (in km/h).

Worked-out example: choose the model

We want to model the **fuel consumption** y (L/100km) with respect to the **car speed** x (in km/h).

What model could you use for the dependency between x and y?

Worked-out example: choose the model

We want to model the **fuel consumption** y (L/100km) with respect to the **car speed** x (in km/h).

What model could you use for the dependency between x and y?

By a simple linear regression model:

$$y = \theta_0 + \theta_1 . x + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Worked-out example: choose the model

We want to model the **fuel consumption** y (L/100km) with respect to the **car speed** x (in km/h).

What model could you use for the dependency between x and y?

By a simple linear regression model:

$$y = \theta_0 + \theta_1.x + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

What is the associated loss?

Loss

Write the negative log-likelihood:

$$\mathcal{L}(\theta, x, y) = \frac{1}{2\sigma^2} \sum_i [y_i - (\theta_0 + \theta_1.x_i)]^2$$

Worked-out example: choose the model

We want to model the **fuel consumption** y (L/100km) with respect to the **car speed** x (in km/h).

What model could you use for the dependency between x and y?

By a simple linear regression model:

$$y = \theta_0 + \theta_1.x + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

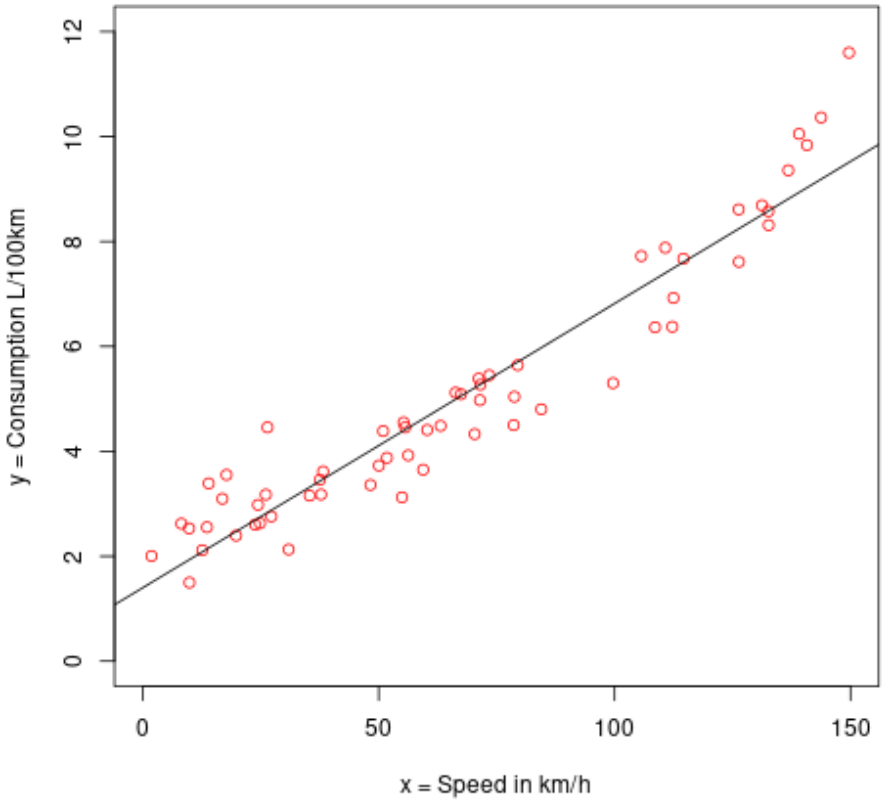
What is the associated loss?

Loss

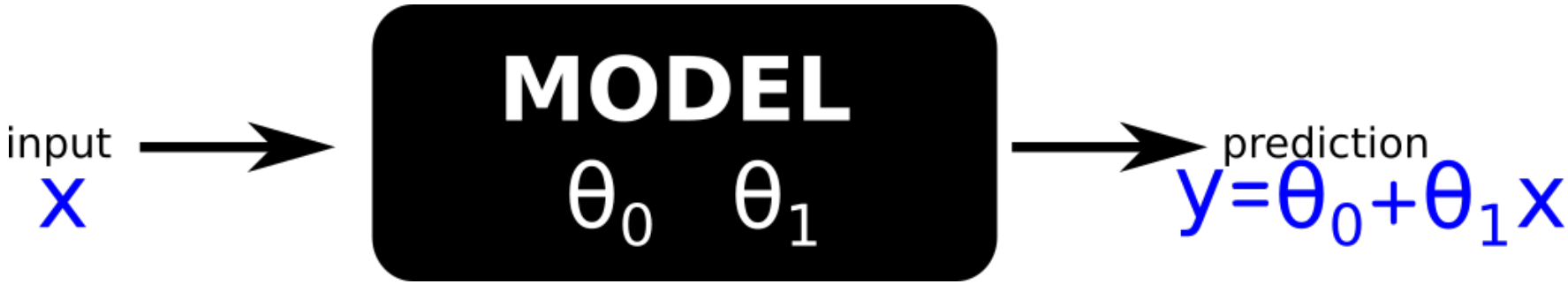
Write the negative log-likelihood:

$$\mathcal{L}(\theta, x, y) = \sum_i [y_i - (\theta_0 + \theta_1.x_i)]^2$$

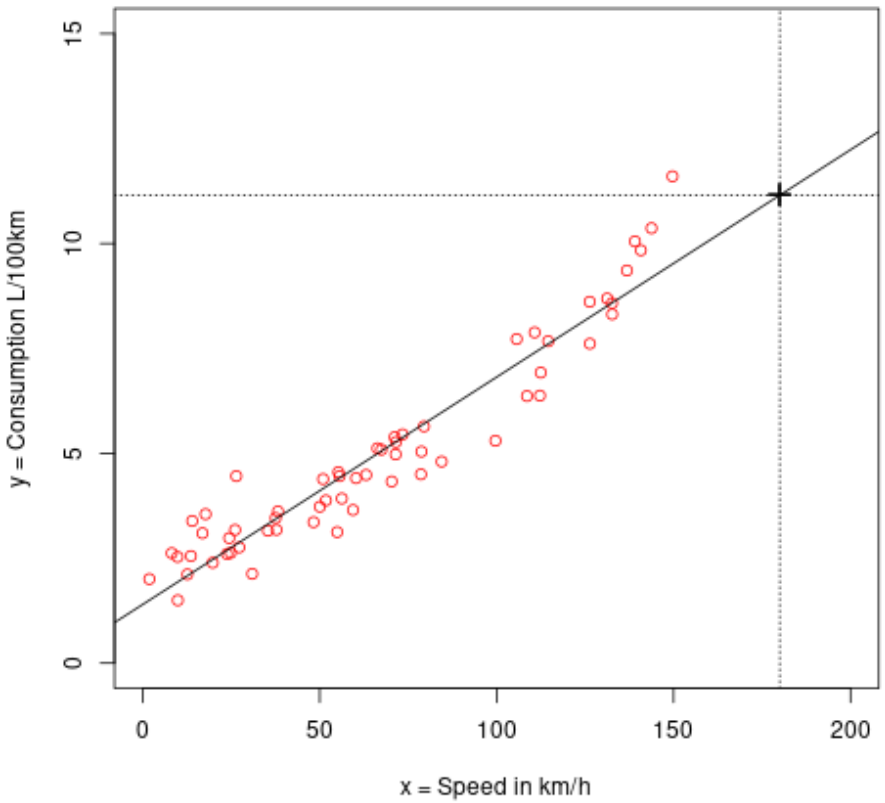
Worked-out example: the fit



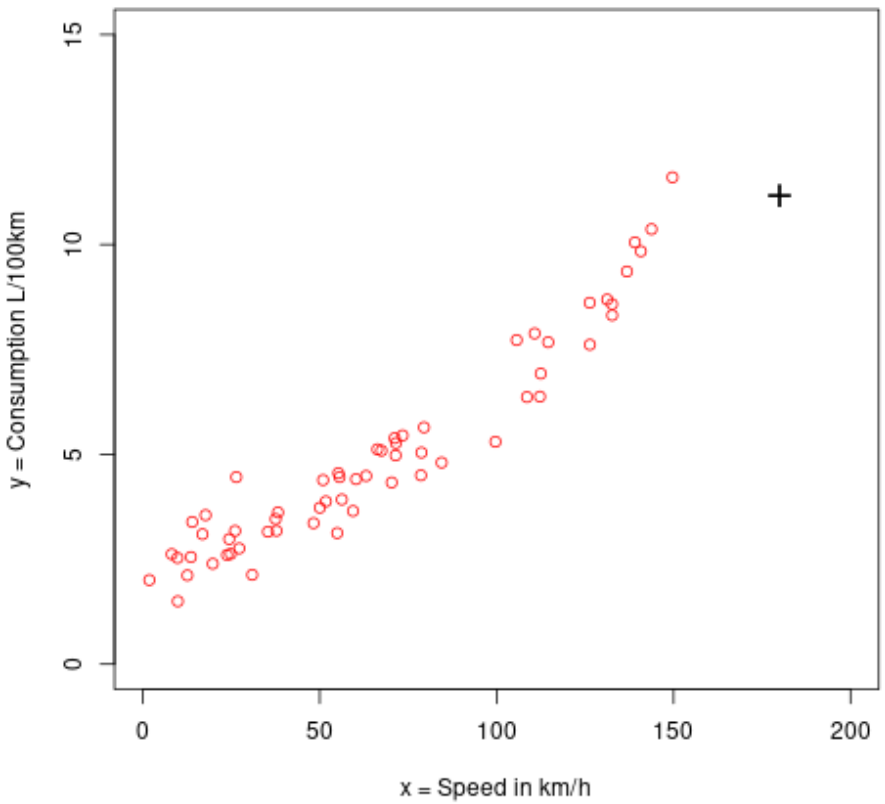
Worked-out example: the prediction



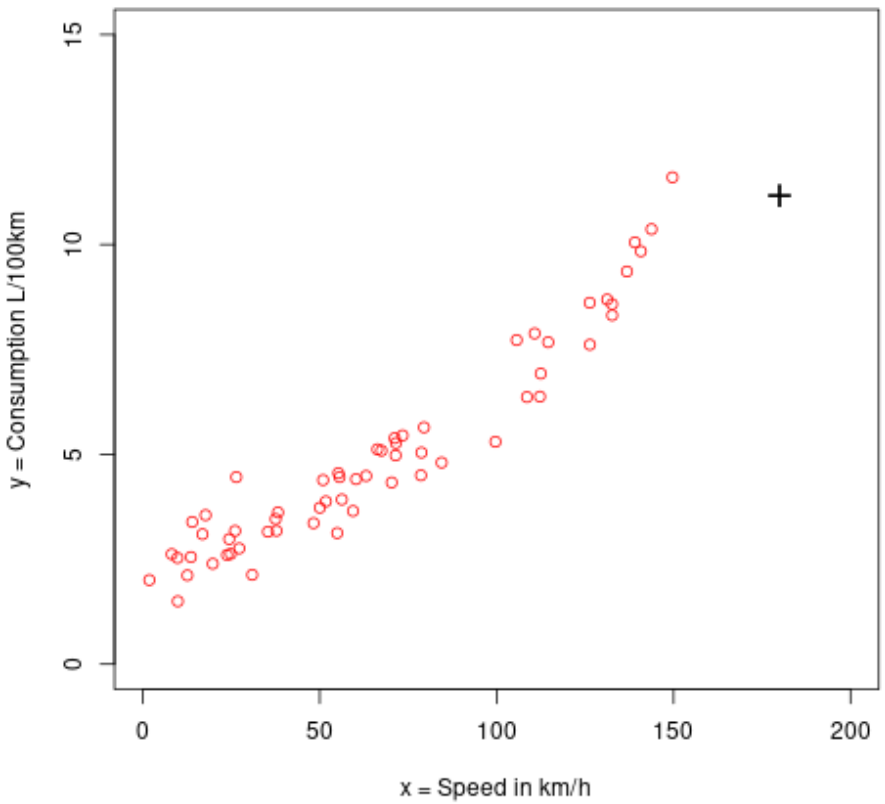
Worked-out example: the prediction



Worked-out example: toward more complex models

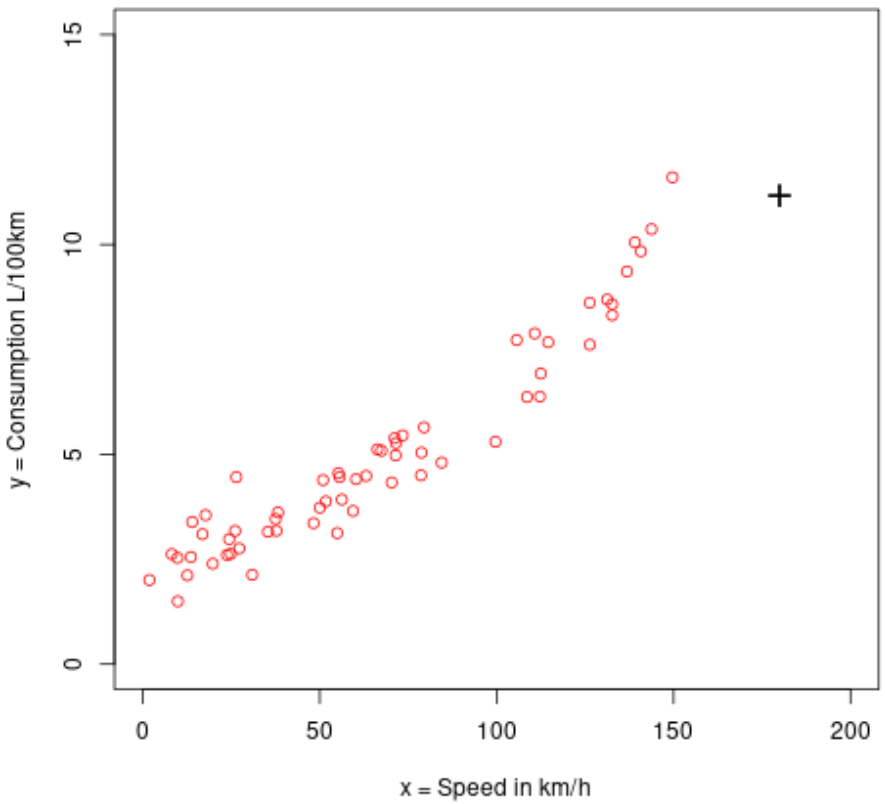


Worked-out example: toward more complex models



Any comment?

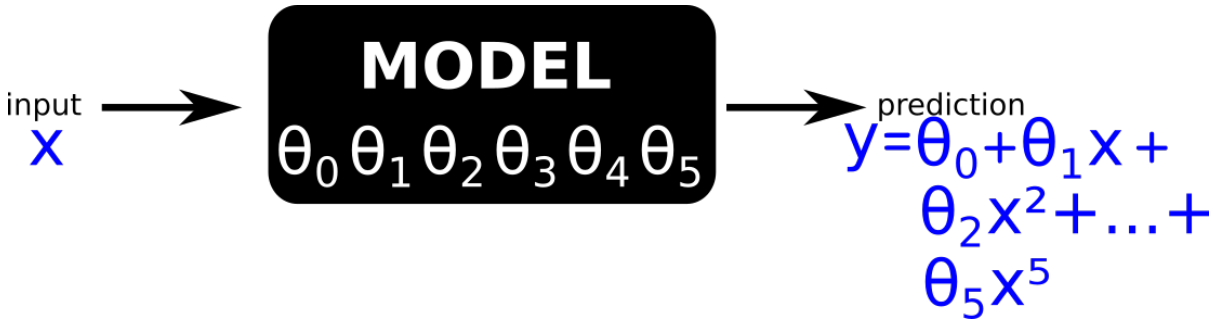
Worked-out example: toward more complex models



Any comment?

This phenomenon is known as **underfitting**

Worked-out example: toward more complex models



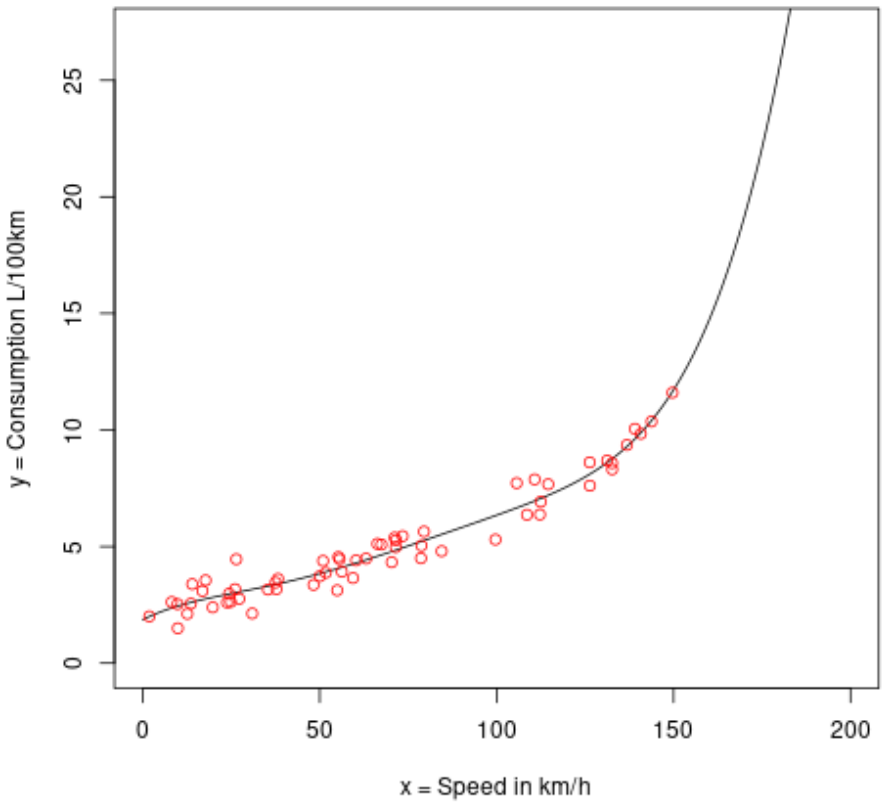
Worked-out example: toward more complex models

$\theta_0, \theta_1 ..., \theta_5$ such that

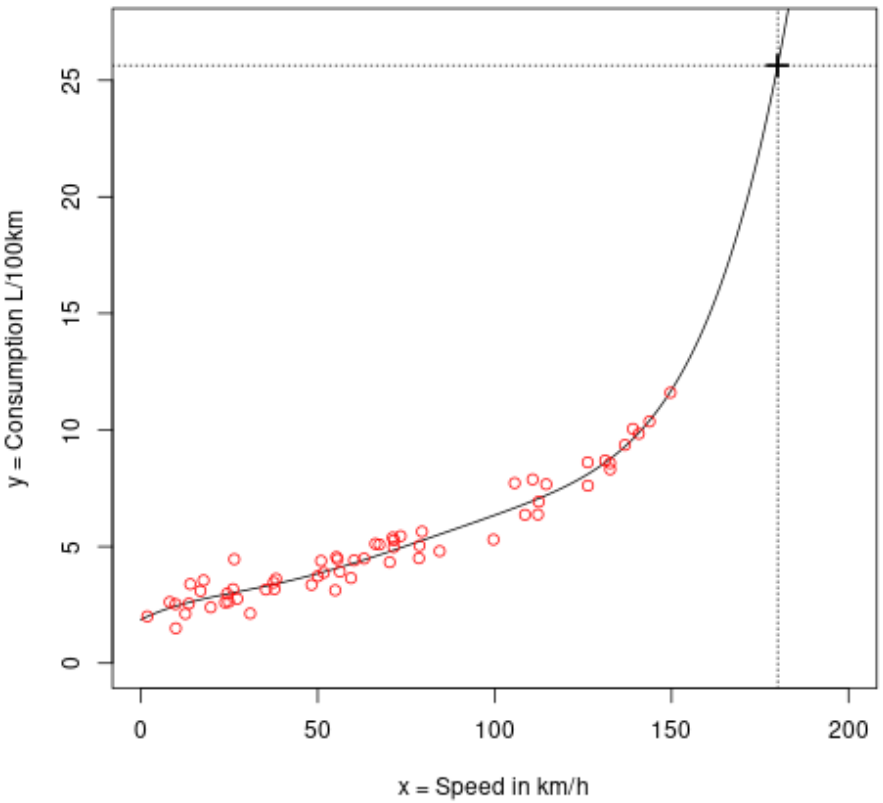
$$\mathcal{L}(\theta_0, ..., \theta_5) = \sum_{i=1}^{N} [y_i - (\sum_{j=0}^{5} \theta_j . x_i^j)]^2 \quad (1)$$

is **minimal**.

Worked-out example: toward more complex models



Worked-out example: toward more complex models



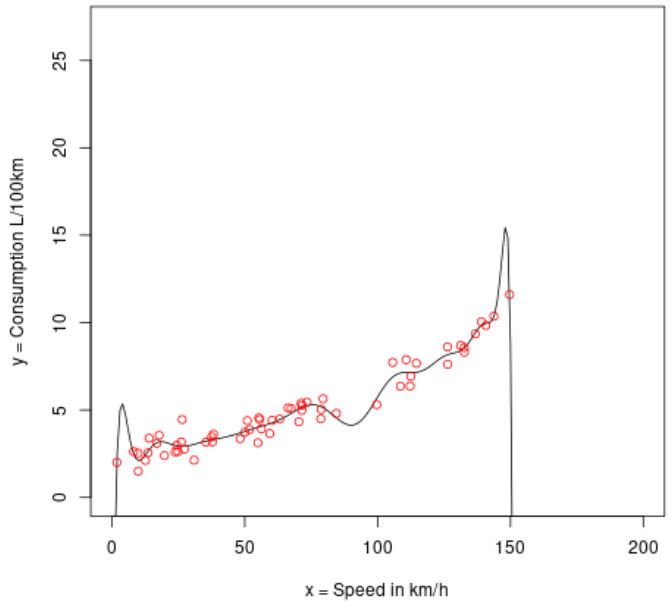
Worked-out example: toward more complex models

Informal definition

We will say that the polynomial model of degree 5 is more **expressive** than the linear regression.

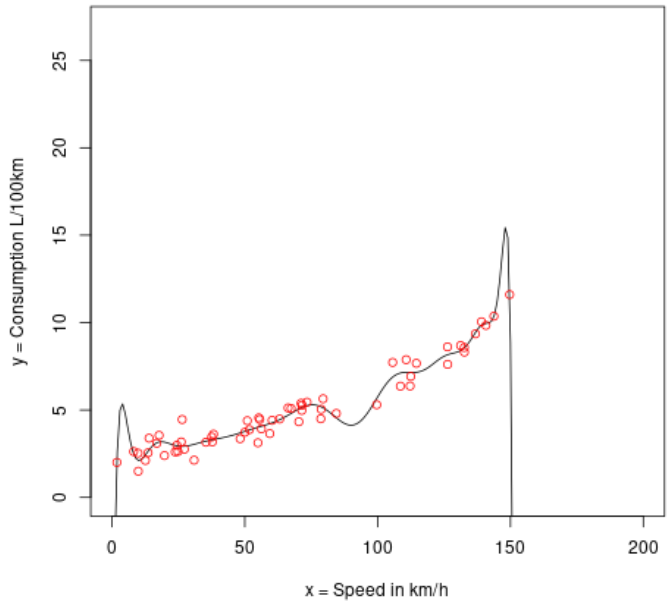
Parameters: the more the better?

Parameters: the more the better?



With 30 parameters: $\theta_0, ...\theta_{29}$

Parameters: the more the better?

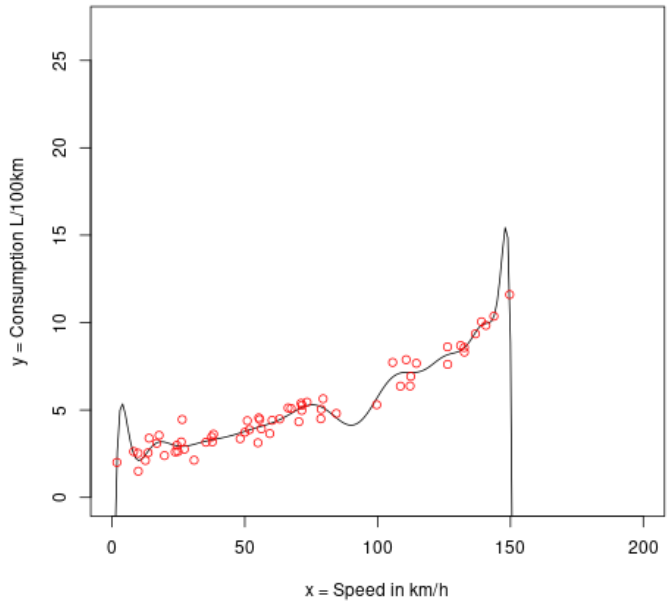


With 30 parameters: $\theta_0, ...\theta_{29}$

Definition

The phenomenon is called **overfit**.

Parameters: the more the better?

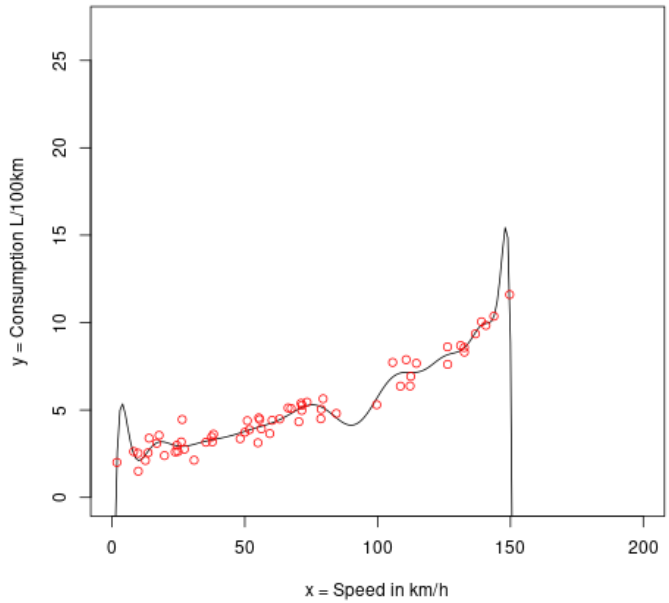


With 30 parameters: $\theta_0, ...\theta_{29}$

Definition

The phenomenon is called **overfit**. Mainly happens because of **hyperparametrization**.

Parameters: the more the better?



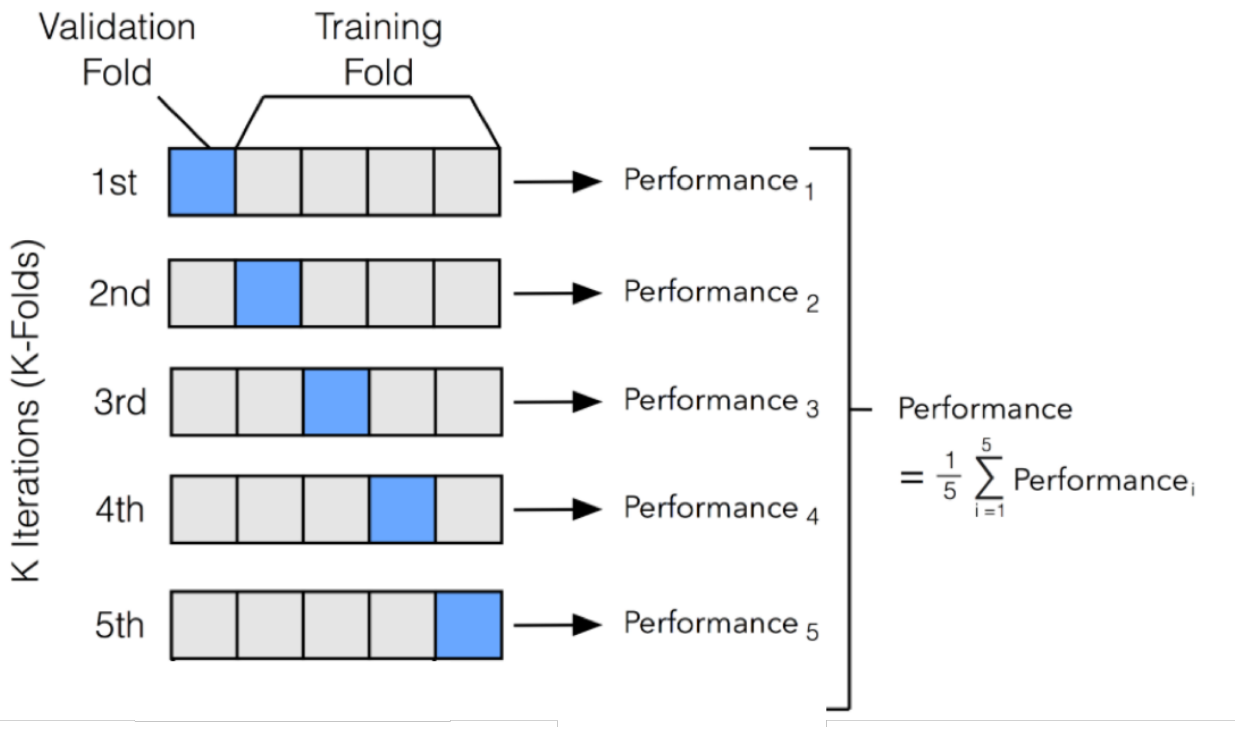
With 30 parameters: $\theta_0, ...\theta_{29}$

Definition

The phenomenon is called **overfit**. Mainly happens because of **hyperparametrization**.

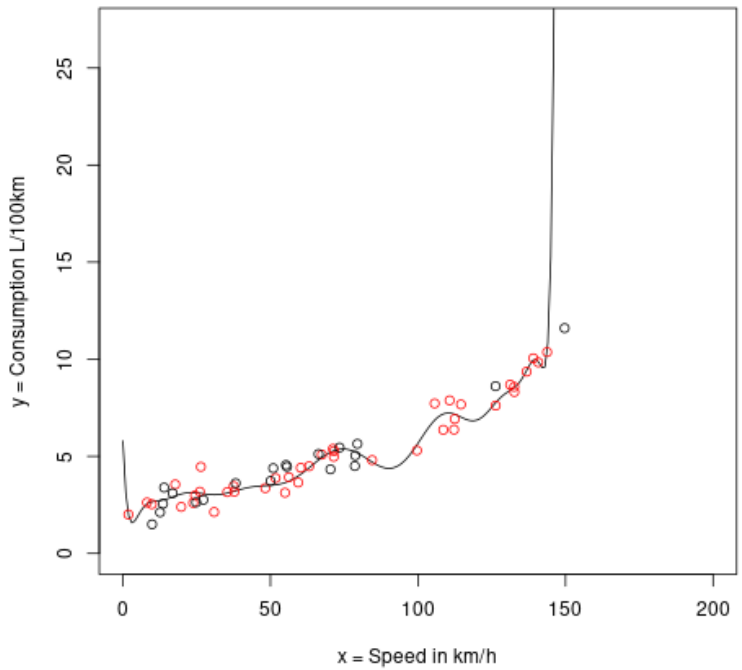
How to control overfit?

Cross-validation to control overfit



Cross-validation example: 30 parameters

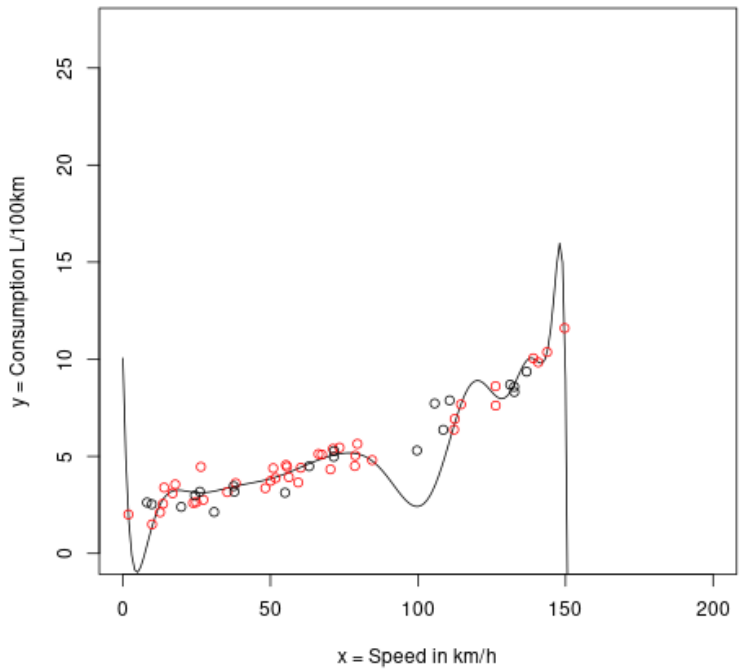
Red: training set Black: validation set



30 parameters, fold 1

Cross-validation example: 30 parameters

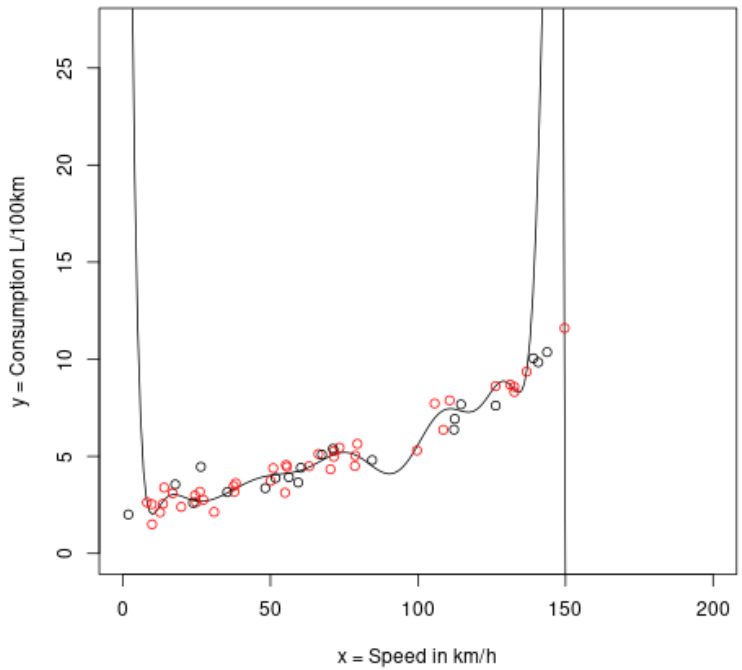
Red: training set Black: validation set



30 parameters, fold 2

Cross-validation example: 30 parameters

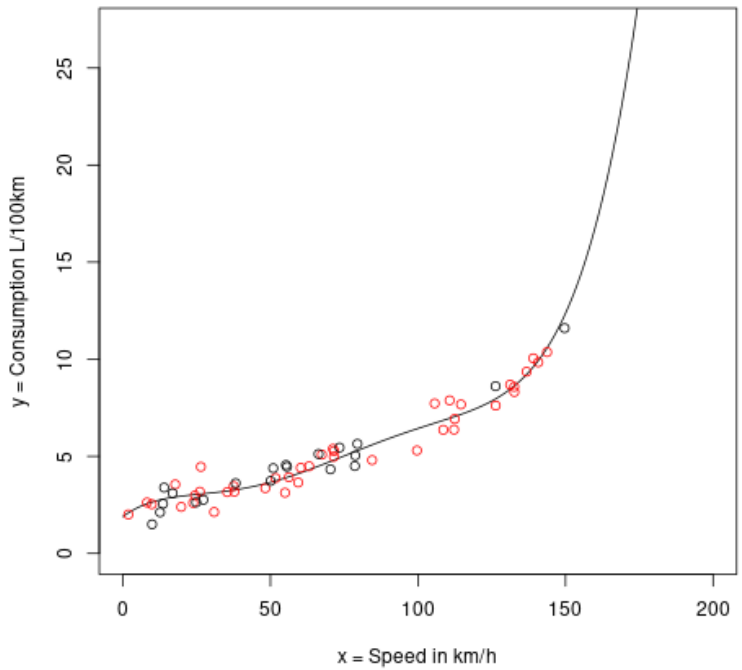
Red: training set Black: validation set



30 parameters, fold 3

Cross-validation example: 6 parameters

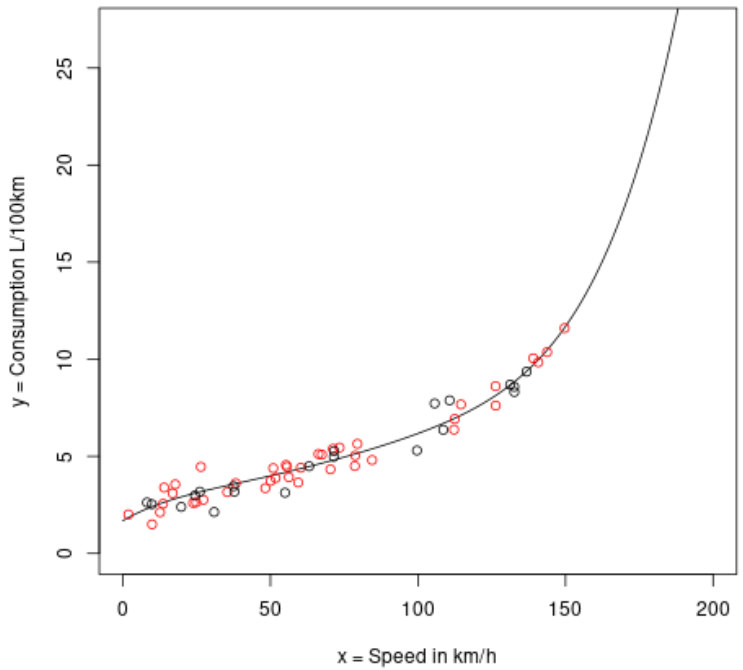
Red: training set Black: validation set



6 parameters, fold 1

Cross-validation example: 6 parameters

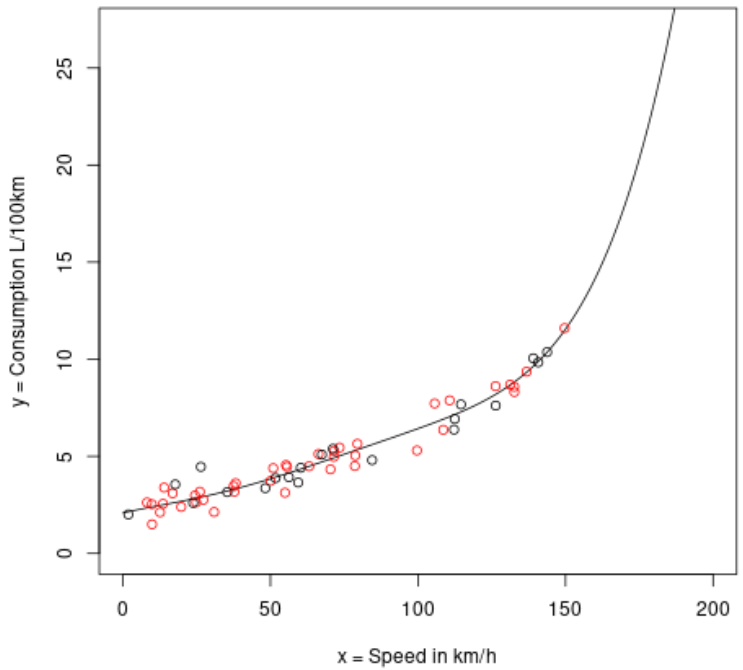
Red: training set Black: validation set



6 parameters, fold 2

Cross-validation example: 6 parameters

Red: training set Black: validation set



6 parameters, fold 3

What do you observe?

- Lower error on training with ___ parameters
- If the error on the validation is much higher than on the training set, it means that the model is ___.
- Naively, a model with ___ parameters will have ___ variance.

What do you observe?

- Lower error on training with **more** parameters
- If the error on the validation is much higher than on the training set, it means that the model is ___.
- Naively, a model with ___ parameters will have ___ variance.

What do you observe?

- Lower error on training with **more** parameters
- If the error on the validation is much higher than on the training set, it means that the model is **overfiting**.
- Naively, a model with ___ parameters will have ___ variance.

What do you observe?

- Lower error on training with **more** parameters
- If the error on the validation is much higher than on the training set, it means that the model is **overfiting**.
- Naively, a model with **more (less)** parameters will have **more (less)** variance.

More specifically, for N training sets $X_i \in X_{train}$ we compute N models f_i and their predictions $f_i(X_{test})$. The expectation of the squared error (mean sq error) of all models $\mathbb{E}[||y - \hat{y}||^2]$ can be decomposed as:

$$\begin{aligned}
\mathbb{E}[||y - \hat{y}||^2] &= \mathbb{E}[||y - \mathbb{E}[\hat{y}] + \mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= \mathbb{E}[||y - \mathbb{E}[\hat{y}]||^2] + 2 \times 0 + \mathbb{E}[||\mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= ||y - \mathbb{E}[\hat{y}]||^2 + \mathbb{E}[||\mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= \text{bias}^2 + \text{variance}
\end{aligned} \tag{1}$$

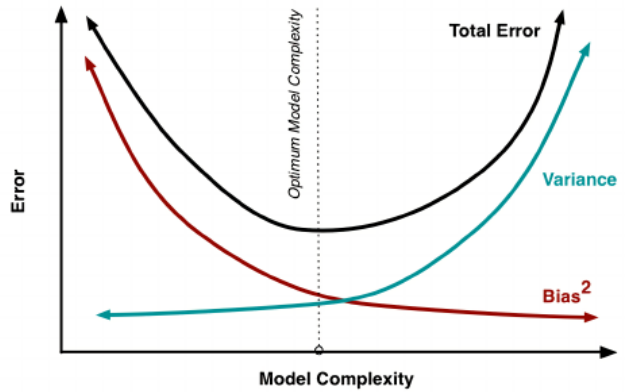
where $\mathbb{E}[\hat{y}]$ is the mean of the predictions of all models.

More specifically, for N training sets $X_i \in X_{train}$ we compute N models f_i and their predictions $f_i(X_{test})$. The expectation of the squared error (mean sq error) of all models $\mathbb{E}[||y - \hat{y}||^2]$ can be decomposed as:

$$\begin{aligned}
\mathbb{E}[||y - \hat{y}||^2] &= \mathbb{E}[||y - \mathbb{E}[\hat{y}] + \mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= \mathbb{E}[||y - \mathbb{E}[\hat{y}]||^2] + 2 \times 0 + \mathbb{E}[||\mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= ||y - \mathbb{E}[\hat{y}]||^2 + \mathbb{E}[||\mathbb{E}[\hat{y}] - \hat{y}||^2] \\
&= \text{bias}^2 + \text{variance}
\end{aligned} \tag{1}$$

where $\mathbb{E}[\hat{y}]$ is the mean of the predict

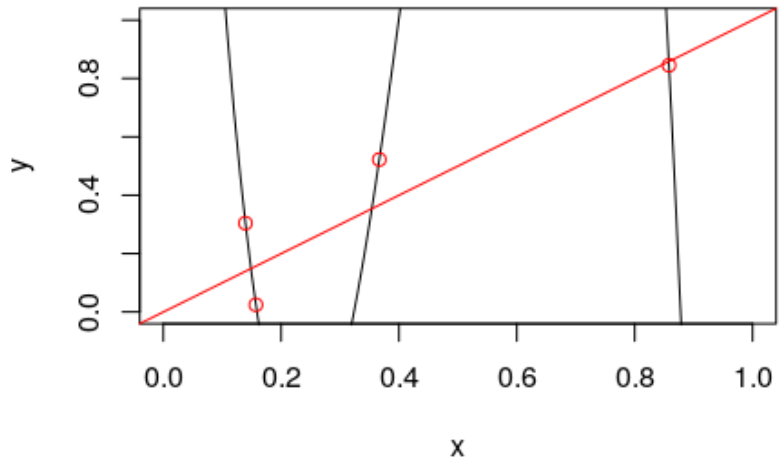
This is known as the **bias-variance trade-off**:



Regularization motivation

Let's come back to the model $y = \sum_{i=0}^{3} \theta_i x^i + \epsilon$.

The max likelihood with 4 points gives a θ fitting perfectly the points:



Maximum *likelihood* coefficients:

θ_0	θ_1	θ_2	θ_3
5.169	-54.388	155.755	-114.487

What makes you think that the model is wrong?

Regularization

The idea of regularization

Definition (well...)

Regularization is a set of methods for avoiding "unrealistic zones" in your parameter space.

Along the tutorials we will use:

- Ridge penalization (avoids high values of parameters)
- Lasso penalization (favors not using some parameters)

Other types of regularization (for Neural Networks in particular) include:

- Gaussian noise (augmenting data)
- Dropout (favors independence in the responsibilities of the parameters)

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.
So we can integrate *a priori* knowledge in our model.

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.
So we can integrate *a priori* knowledge in our model.

We consider $\theta_0, ...\theta_3$ as random variables (i.e. quantity having uncertainties).
We *model them*, for example with normal distributions centered on likely values (e.g. $\mu_0 = 0.1$, $\mu_1 = ...$) with some likely variability (e.g. $\eta_0 = 0.005$, etc.).

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.
So we can integrate *a priori* knowledge in our model.

We consider $\theta_0, ...\theta_3$ as random variables (i.e. quantity having uncertainties).
We *model them*, for example with normal distributions centered on likely values (e.g. $\mu_0 = 0.1$, $\mu_1 = ...$) with some likely variability (e.g. $\eta_0 = 0.005$, etc.).

The model becomes:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta_i \sim \mathcal{N}(\mu_i, \eta_i^2)$$
$$y = \sum \theta_i . x^i + \epsilon$$

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.
So we can integrate *a priori* knowledge in our model.

We consider $\theta_0, ...\theta_3$ as random variables (i.e. quantity having uncertainties).

We *model them*, for example with normal distributions centered on likely values (e.g. $\mu_0 = 0.1$, $\mu_1 = ...$) with some likely variability (e.g. $\eta_0 = 0.005$, etc.).

The model becomes:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta_i \sim \mathcal{N}(\mu_i, \eta_i^2)$$
$$y = \sum \theta_i . x^i + \epsilon$$

What is "random" here?

Prior distributions

In the Bayesian world, probabilities represent the degree of knowledge.
So we can integrate *a priori* knowledge in our model.

We consider $\theta_0, ...\theta_3$ as random variables (i.e. quantity having uncertainties).

We *model them*, for example with normal distributions centered on likely values (e.g. $\mu_0 = 0.1$, $\mu_1 = ...$) with some likely variability (e.g. $\eta_0 = 0.005$, etc.).

The model becomes:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta_i \sim \mathcal{N}(\mu_i, \eta_i^2)$$
$$y = \sum \theta_i . x^i + \epsilon$$

What is "random" here?

The θ_i are model **parameters** (inferred from the training data).
The μ_i and η_i are **hyperparameters** (not inferred from the training).

Worked out example

Consider a simple model:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta \sim \mathcal{N}(0, \eta^2)$$
$$y = \theta x + \epsilon$$

Worked out example

Consider a simple model:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta \sim \mathcal{N}(0, \eta^2)$$
$$y = \theta x + \epsilon$$

Exercise

1. Compute the posterior probability distribution

$$p(\theta|y, x) \propto p(y|\theta, x).p(\theta)$$

Worked out example

Consider a simple model:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta \sim \mathcal{N}(0, \eta^2)$$
$$y = \theta x + \epsilon$$

Exercise

1. Compute the posterior probability distribution

$$p(\theta|y, x) \propto p(y|\theta, x).p(\theta)$$

2. Show that maximizing the posterior probability distribution is the same as solving the following optimization problem:

$$\arg\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_2^2$$

Worked out example

Consider a simple model:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\theta \sim \mathcal{N}(0, \eta^2)$$
$$y = \theta x + \epsilon$$

Exercise

1. Compute the posterior probability distribution

$$p(\theta|y, x) \propto p(y|\theta, x).p(\theta)$$

2. Show that maximizing the posterior probability distribution is the same as solving the following optimization problem:

$$\arg\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_2^2$$

3. What it is the value of λ?

Toward Ridge regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2$$

Toward Ridge regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_2^2$$

Toward Ridge regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_2^2$$

This is called **Ridge regularization**.

What is it enforcing?

Toward Ridge regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_2^2$$

This is called **Ridge regularization**.

What is it enforcing?

It tells the model **to avoid high values** for the parameters.

General justification of Ridge regularization

Complexity

The complexity of a model is the dimensionality of the space it can describe, usually linked to the number of parameters.

A model with p binary parameters θ_i can describe ? outputs.

General justification of Ridge regularization

Complexity

The complexity of a model is the dimensionality of the space it can describe, usually linked to the number of parameters.

A model with p binary parameters θ_i can describe 2^p outputs.

General justification of Ridge regularization

Complexity

The complexity of a model is the dimensionality of the space it can describe, usually linked to the number of parameters.

A model with p binary parameters θ_i can describe 2^p outputs.

How would you measure that for continuous parameters?

General justification of Ridge regularization

Complexity

The complexity of a model is the dimensionality of the space it can describe, usually linked to the number of parameters.

A model with p binary parameters θ_i can describe 2^p outputs.

How would you measure that for continuous parameters?
With the volume:

$$V_p(r) = K_p.r^p,$$

where r is the radius where the parameters live and K_p a constant associated with the number of parameters.

General justification of Ridge regularization

Complexity

The complexity of a model is the dimensionality of the space it can describe, usually linked to the number of parameters.

A model with p binary parameters θ_i can describe 2^p outputs.

How would you measure that for continuous parameters?
With the volume:

$$V_p(r) = K_p.r^p,$$

where r is the radius where the parameters live and K_p a constant associated with the number of parameters.

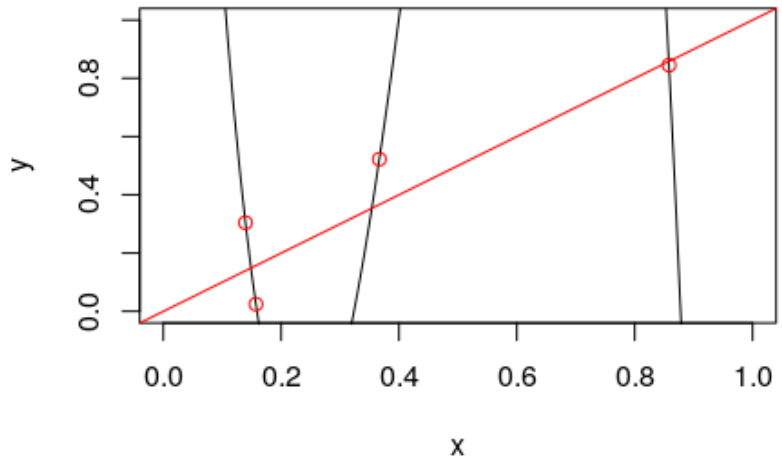
High dimension

In high dimension, there are "more" possible model outputs when parameters have high values.

Ridge regularization example

Let's come back to the model $Y = \sum_{i=0}^{3} \theta_i x^i + \epsilon$.

The maximum likelihood with 4 points will give a θ fitting perfectly the points:



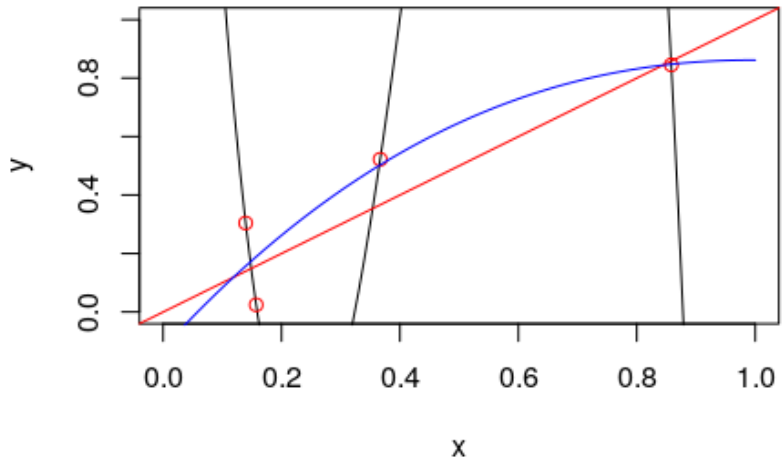
Maximum *likelihood* coefficients:

θ_0	θ_1	θ_2	θ_3
5.169	-54.388	155.755	-114.487

Ridge regularization example

Let's come back to the model $Y = \sum_{i=0}^{3} \theta_i x^i + \epsilon$.

With a prior $\mathcal{N}(0, \eta^2)$ the maximum a posteriori of the vector θ corresponds to (blue curve):



Maximum *a posteriori* coefficients

θ_0	θ_1	θ_2	θ_3
-0.1279	2.2561	-1.5779	0.3180

Sparse learning

Patient	Status	SNP1	SNP2	SNP3	SNP4	...
A	**0**	0	0	0	1	...
B	**1**	1	0	0	1	...
C	**1**	1	0	0	0	...
...						

From Ridge to Lasso

Suppose you model a variable Y depending on some explanatory variables x with a linear model:

$$Y = \theta_0 + \sum_{i=1}^{p} \theta_i . x_i + \epsilon$$

Imagine now that you know that actually **only few** variables actually explain your target variable.

Question

A Gaussian prior on θ_i centered on 0 avoids high values of θ_i.
Will this prior push the non-explanatory variables down to 0?

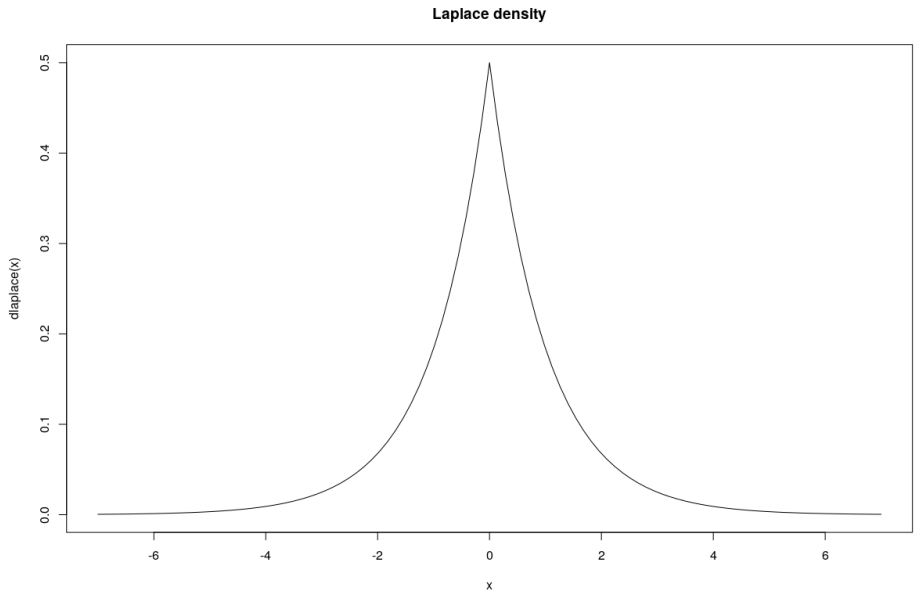
- Think individually - Draw - Rethink (5')
- Vote

Lasso penalization

What should be the shape around 0 of the prior distribution if we want to use less parameters?

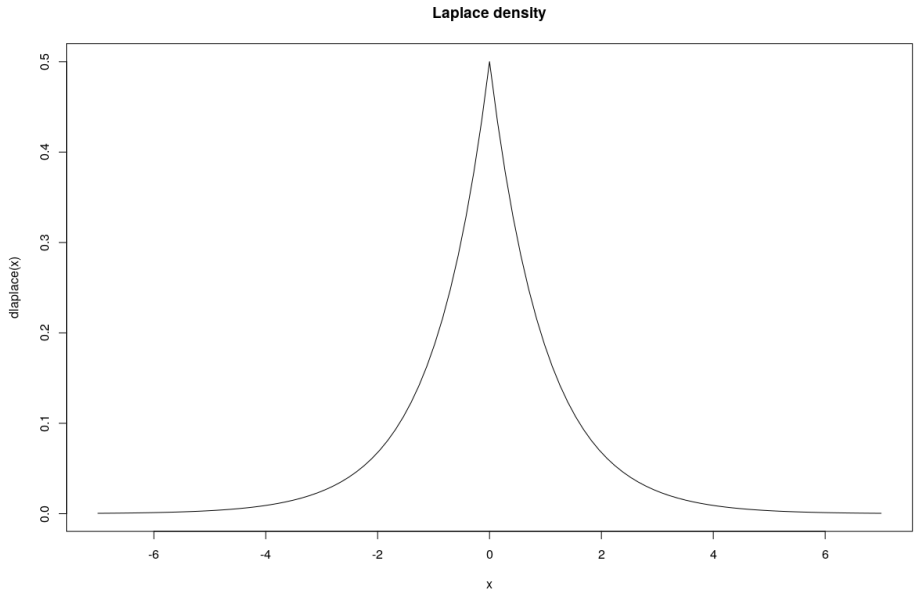
Lasso penalization

What should be the shape around 0 of the prior distribution if we want to use less parameters?
Something like the Laplace density:



$$f(x|\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

Lasso penalization



$$f(x|\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

Exercise

Compute the loss associated to a zero centered Laplace prior distribution.

Toward Lasso regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2$$

Toward Lasso regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_1$$

Toward Lasso regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_1$$

This is called **Lasso regularization**.

What is it enforcing?

Toward Lasso regularization

$$\min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 \quad \rightarrow \quad \min_{\theta} \sum_{i=0}^{N} (y_i - \theta.x_i)^2 + \lambda ||\theta||_1$$

This is called **Lasso regularization**.

What is it enforcing?

It tells the model **to use as few parameters** as possible.

The idea of regularization

Definition (well...)

Regularization is a set of methods for avoiding "unrealistic zones" in your parameter space.

We saw:

- Ridge penalization (avoids high values of parameters)
- Lasso penalization (favors not using some parameters)

Other types of regularization (for Neural Networks in particular) include:

- Gaussian noise (augmenting data)
- Dropout (favors independence in the responsibilities of the parameters)