UPPSALA UNIVERSITET

# Statistical Machine Learning

*Lecture 2 – Linear regression, regularization*

UPPSALA UNIVERSITET

**Sebastian Mair**
`https://smair.github.io/`
Department of Information Technology
Uppsala University

Course webpage

UPPSALA UNIVERSITET

# Summary of Lecture 1 (I/II)

**What is this course about? Supervised machine learning**

In one sentence:

Methods for automatically learning (training, estimating, ...) **a model** for the relationship between

- the **input** $\mathbf{x}$, and
- the **output** $y$

from observed **training data**

$$\mathcal{T} := \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}.$$

# Summary of Lecture 1 (II/II)

Regression vs. classification

- **Numerical** variables take on numerical values (real numbers, integer values, ...).
- **Categorical** variables take on values in one of $K$ distinct classes, e.g. "true or false", "disease type $A$, $B$ or $C$".

**Regression** is when the output $y$ is numerical.

**Classification** is when the output $y$ is categorical.

# Summary of Lecture 1 (II/II)

What maths do we need.

- **Calculus** Finding a parameter which minimizes the distance between two points.
- **Matrix algebra** For keeping track of sum of squares.
- **Probability theory** Estimating parameters. The normal distribution important.

One key idea that brings these together is maximum likelihood. Finding the model that is maximally likely (closest to data).

# Outline – Lecture 2

**Aim:** To introduce linear regression and its regularized version.

**Outline:**

1. Summary of Lecture 1
2. Linear regression models
3. Maximum likelihood and least squares
4. Regularization
   - Ridge regression
   - LASSO

---

*Linear regression is the foundation of statistics and (supervised) machine learning.*

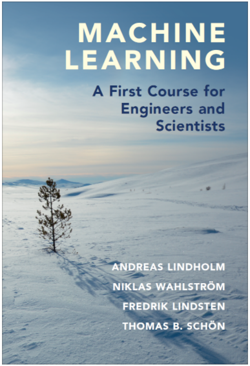
# The course book

This course was developed over the last years.

As a result, a book based on these notes was published as a textbook with Cambridge University Press.

Book website: `smlbook.org`

Feedback can be provided via the Github page.




**Please read as part of your studies.**

# Regression

- **Input variable** $\mathbf{x}$
- **Output variable** $y$

**Regression:** Learning a model explaining $y$ from $\mathbf{x}$, when $y$ is numerical, i.e.,

$$y = f(\mathbf{x}; \beta) + \epsilon.$$

Here, $\beta$ are the **parameters** of the model.

($y$ categorical $\rightarrow$ classification)

# Numerical or categorical variables

**Numerical or categorical?**

17.31 kg, 22.37 kg, 51.34 kg
1 = brown hair, 2 = red hair, 3 = blonde hair
Adenine, Thymine, Cytosine, Guanine
1 bike, 2 bikes, 5 bikes

# Numerical or categorical variables

**Numerical or categorical?**

| | |
|---|---|
| 17.31 kg, 22.37 kg, 51.34 kg | Numerical |
| 1 = brown hair, 2 = red hair, 3 = blonde hair | |
| Adenine, Thymine, Cytosine, Guanine | |
| 1 bike, 2 bikes, 5 bikes | |

# Numerical or categorical variables

**Numerical or categorical?**

| | |
|---|---|
| 17.31 kg, 22.37 kg, 51.34 kg | Numerical |
| 1 = brown hair, 2 = red hair, 3 = blonde hair | Categorical |
| Adenine, Thymine, Cytosine, Guanine | |
| 1 bike, 2 bikes, 5 bikes | |

# Numerical or categorical variables

**Numerical or categorical?**

| | |
|---|---|
| 17.31 kg, 22.37 kg, 51.34 kg | Numerical |
| 1 = brown hair, 2 = red hair, 3 = blonde hair | Categorical |
| Adenine, Thymine, Cytosine, Guanine | Categorical |
| 1 bike, 2 bikes, 5 bikes | |

# Numerical or categorical variables

**Numerical or categorical?**

| | |
|---|---|
| 17.31 kg, 22.37 kg, 51.34 kg | Numerical |
| 1 = brown hair, 2 = red hair, 3 = blonde hair | Categorical |
| Adenine, Thymine, Cytosine, Guanine | Categorical |
| 1 bike, 2 bikes, 5 bikes | Numerical |

UPPSALA UNIVERSITET

# Numerical or categorical variables

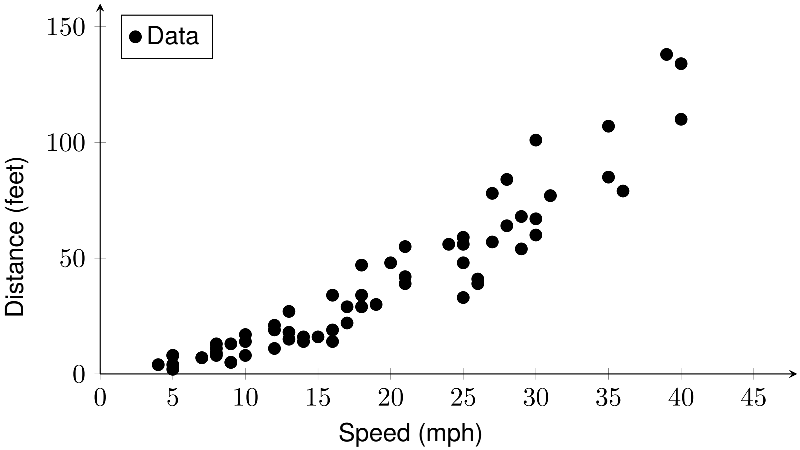
**Numerical or categorical?**

| | |
|---|---|
| 17.31 kg, 22.37 kg, 51.34 kg | Numerical |
| 1 = brown hair, 2 = red hair, 3 = blonde hair | Categorical |
| Adenine, Thymine, Cytosine, Guanine | Categorical |
| 1 bike, 2 bikes, 5 bikes | Numerical |

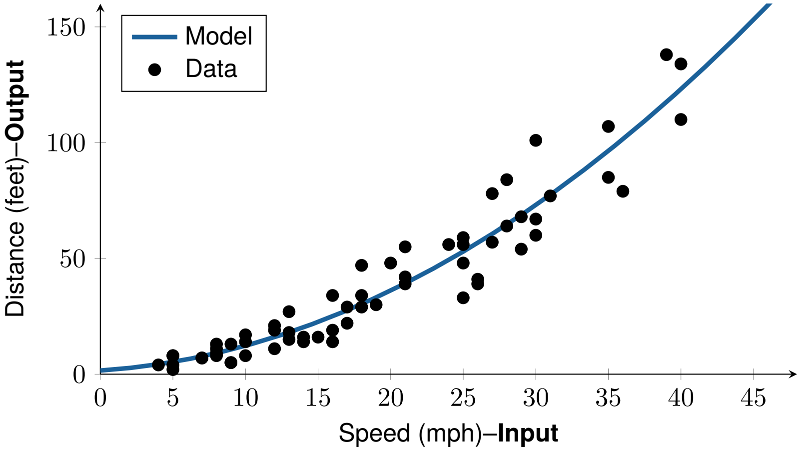
Categorical output variable? $\rightarrow$ classification. (But can be done with regression!)

Categorical input variable? Still regression!

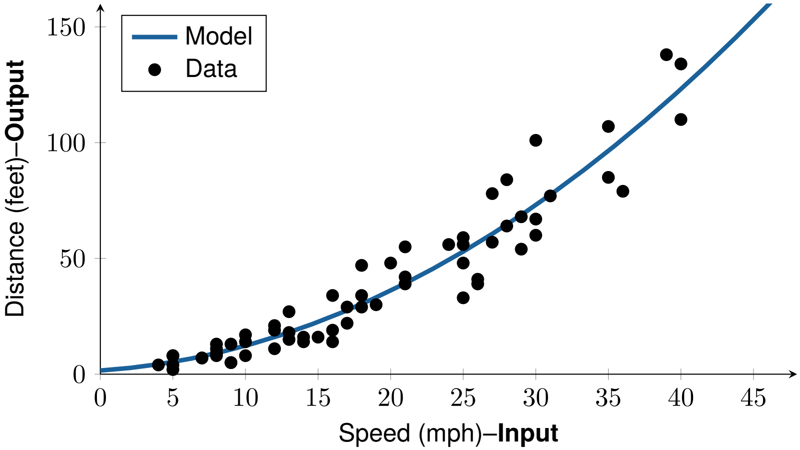
# Regression example: car stopping distances



# Regression example: car stopping distances



# Regression example: car stopping distances




*(in fact a linear regression model with nonlinear transformation of the input variables)*

# Regression example: Alpha Go zero



UPPSALA UNIVERSITET

# Regression example: Alpha Go zero


- Input: State of the game ($19 \times 19$ grid, either black, white or blank)
- Output: Probability for the current player to win the game

\+ *reinforcement learning*

Silver et al. **Mastering the game of Go with deep neural networks and tree search**, *Nature* 529, 484–489, 2016.

UPPSALA UNIVERSITET

# Regression example: Alpha Go zero


- Input: State of the game ($19 \times 19$ grid, either black, white or blank)
- Output: Probability for the current player to win the game

\+ *reinforcement learning*

Silver et al. **Mastering the game of Go with deep neural networks and tree search**, *Nature* 529, 484–489, 2016.

- Input: Same
- Output: Probability for the current player to win the game *and* what move to make

Silver et al. **Mastering the game of Go without human knowledge**, *Nature* 550, 354–359, 2017.

Silver et al. **A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play**, *Science*, 362(6419): 1140–1144, 2018.

# Linear regression

*"Linear regression = Regression with a linear model",*

Output $y$ is linear combination of $k$ inputs $x_1, \dots, x_k$ plus some noise/error $\epsilon$,

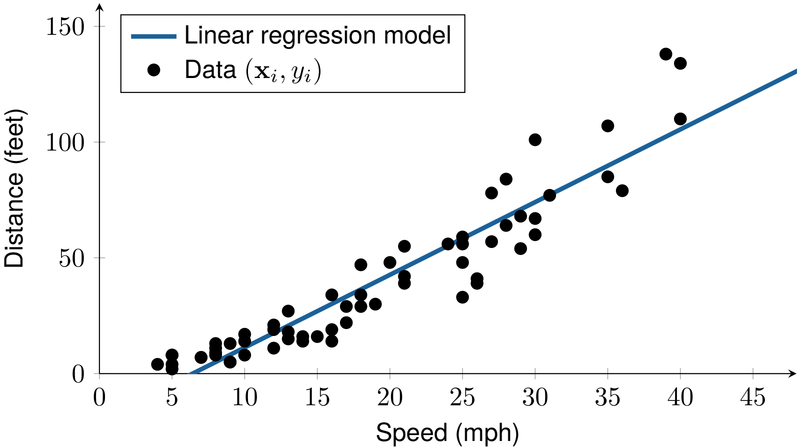
$$y = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k}_{f(\mathbf{x};\beta)} + \epsilon.$$

*Workflow (for most methods, not only linear regression):*

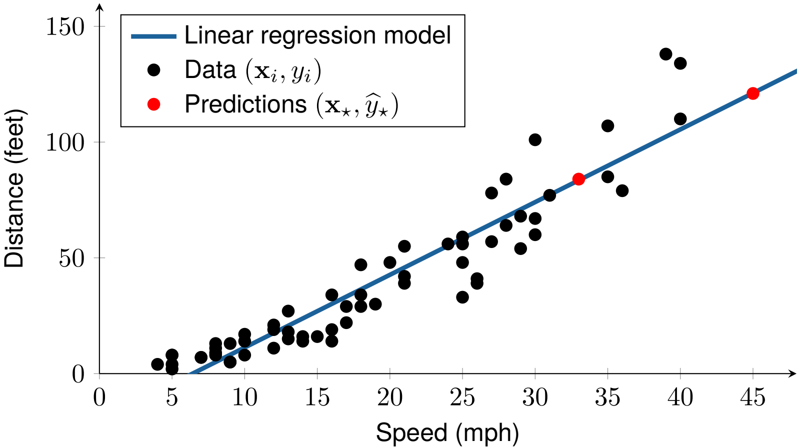
1. Learn/train/estimate model from training data $\mathcal{T}$: find $\widehat{\beta}_0, \widehat{\beta}_1, \dots, \widehat{\beta}_k$
2. Predict output for new test input $\mathbf{x}_\star$ using the model $\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x_{\star 1} + \widehat{\beta}_2 x_{\star 2} + \cdots + \widehat{\beta}_k x_{\star k}$

UPPSALA UNIVERSITET

# Linear regression ($k = 1$)



# Linear regression ($k = 1$)



# Learning the model from data

Linear regression model:

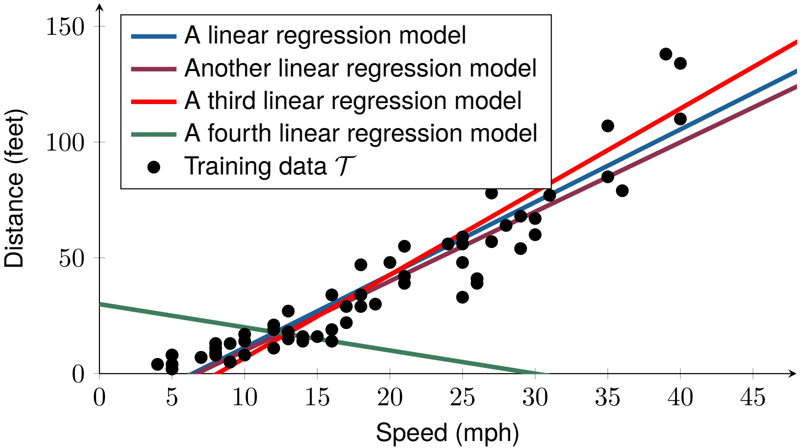
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon$$

How to choose $\beta_0, \beta_1, \ldots, \beta_k$ (=$\beta$, column vector)?

Use **training data** $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$!

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ik} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & -\mathbf{x}_1^\mathsf{T}- \\ 1 & -\mathbf{x}_2^\mathsf{T}- \\ \vdots & \vdots \\ 1 & -\mathbf{x}_n^\mathsf{T}- \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

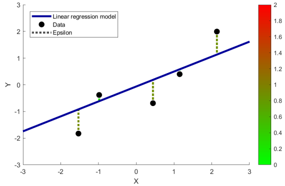
# What is a good model?



UPPSALA UNIVERSITET

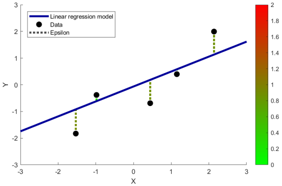
# Learning using maximum likelihood

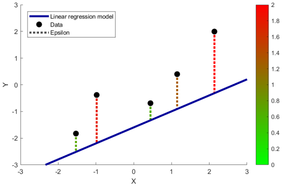
Learning a model from data is a matter of looking at the errors $\epsilon$!

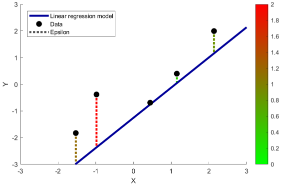


# Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors $\epsilon$!

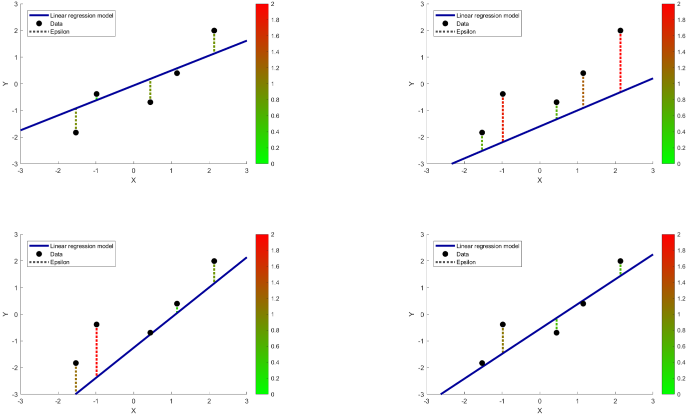


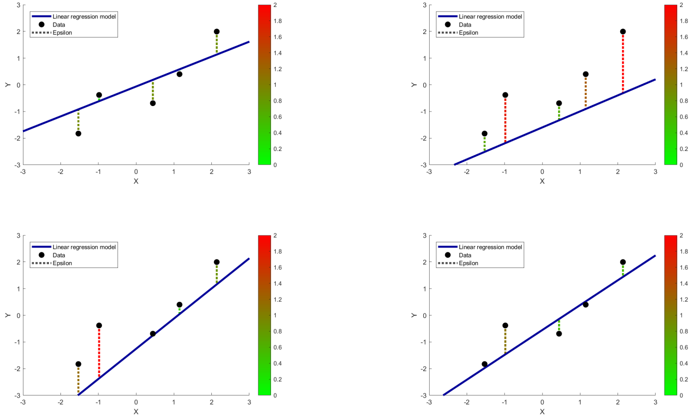
# Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors $\epsilon$!



# Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors $\epsilon$!




**Maximum likelihood**: Think of $\epsilon$ (dotted) as random variables, and ***choose the model*** (solid) ***such that the resulting $\epsilon$ are as likely as possible***.

# Linear regression model in matrix form

UPPSALA UNIVERSITET

Recall our linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 I).$$

Assumptions (for now):

1. $\mathbf{y}$ – observed **random** variable.
2. $\boldsymbol{\beta}$ – unknown **deterministic** variable.
3. $\boldsymbol{X}$ – known **deterministic** variable.
4. $\boldsymbol{\epsilon}$ – unknown **random** variable.
5. $\sigma_\epsilon$ – unknown/known **deterministic** variable.

# Learning using maximum likelihood

Using the **maximum likelihood principle**

$$\widehat{\beta} = \underset{\beta}{\text{argmax}}\; p(\mathbf{y} \mid \mathbf{X}; \beta)$$

and assuming $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ independently for each data point $i$

$$\Rightarrow p(y_i|\mathbf{x}_i;\beta) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} - y_i)^2\right)$$

$$\Rightarrow p(\mathbf{y}|\mathbf{X};\beta) = \prod_{i=1}^{n} p(y_i|x_i;\beta) \propto \exp\left(-\frac{1}{2\sigma_\epsilon^2}\sum_{i=1}^{n}(\beta_0 + \cdots + \beta_k x_{ik} - y_i)^2\right)$$

$$\Rightarrow \widehat{\beta} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} - y_i)^2 = \underset{\beta}{\text{argmin}}\; \underbrace{\|\mathbf{X}\beta - \mathbf{y}\|_2^2}_{\text{Loss function induced by maximum likelihood}},$$

the **least squares** problem is achieved.

# Least squares in matrix form

The least squares problem is given by

$$\widehat{\beta} = \underset{\beta}{\text{argmin}} \, \|\mathbf{X}\beta - \mathbf{y}\|_2^2.$$

UPPSALA UNIVERSITET

# Least squares in matrix form

The objective of the least squares problem can be written as

$$V(\beta) = \|\mathbf{X}\beta - \mathbf{y}\|_2^2 = \beta^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\beta - 2\mathbf{y}^\mathsf{T}\mathbf{X}\beta + \mathbf{y}^\mathsf{T}\mathbf{y}.$$

Minimize by differentiating and setting

$$\frac{\partial V(\beta)}{\partial \beta} = 2\mathbf{X}^\mathsf{T}\mathbf{X}\beta - 2\mathbf{y}^\mathsf{T}\mathbf{X}$$

to zero. Therefore,

$$\mathbf{X}^\mathsf{T}\mathbf{X}\widehat{\beta} = \mathbf{X}^\mathsf{T}\mathbf{y}.$$

UPPSALA UNIVERSITET

# Least squares in matrix form

The least squares problem

$$\widehat{\beta} = \underset{\beta}{\text{argmin}} \, \|\mathbf{X}\beta - \mathbf{y}\|_2^2$$

is solved by the **normal equations**

$$\widehat{\beta} = \left(\mathbf{X}^\mathsf{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}.$$

Remember $\mathbf{X}^\mathsf{T}\mathbf{X}$ is like sum of squares (similar to co-variances of input variables) and $\mathbf{X}^\mathsf{T}\mathbf{y}$ is similar to co-variance between input and output.
For $k = 1$:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

# Linear regression: the key concepts

**The linear regression model**

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \epsilon$$

+

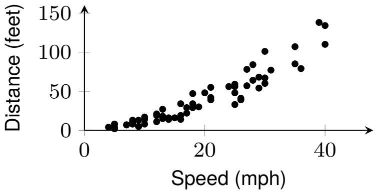
**Maximum likelihood**

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \text{ iid}$$

**Our first learning tool**

# Example

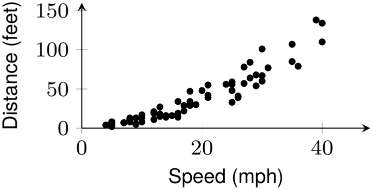
- $x$ = Speed
- $y$ = Distance



# Example

- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \ \ \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$



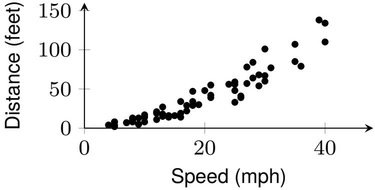
# Example

- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 7 \\ 1 & 7 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 39 \\ 1 & 39 \\ 1 & 40 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$



# Example

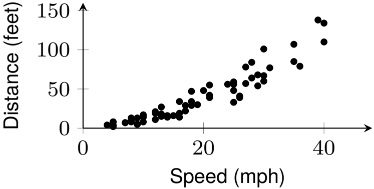
- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 7 \\ 1 & 7 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 39 \\ 1 & 39 \\ 1 & 40 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$

The normal equations $\Rightarrow \widehat{\beta} = \begin{bmatrix} -20.1 \\ 3.1 \end{bmatrix}$



UPPSALA UNIVERSITET

# Example

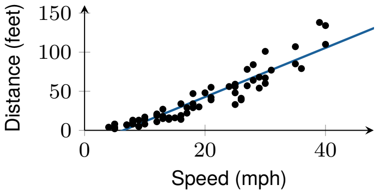
- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 7 \\ 1 & 7 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 39 \\ 1 & 39 \\ 1 & 40 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$

The normal equations $\Rightarrow \widehat{\beta} = \begin{bmatrix} -20.1 \\ 3.1 \end{bmatrix}$



# Example

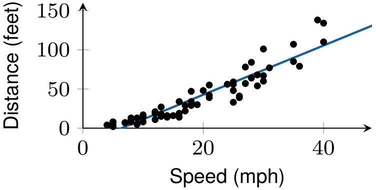
- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 7 \\ 1 & 7 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 39 \\ 1 & 39 \\ 1 & 40 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$

The normal equations $\Rightarrow \widehat{\beta} = \begin{bmatrix} -20.1 \\ 3.1 \end{bmatrix}$




Use the model for predictions!

# Transforming the inputs

*"If the speed $v$ is an input variable, why can't the kinetic energy ($\propto v^2$) be an input variable?"*

**We can make arbitrary nonlinear transformations to the input variables!**

The model is still a linear regression model, since

$$y = \beta_0 + \beta_1 v + \beta_2 v^2 + \beta_3 \cos(v) + \beta_4 \text{arctan}(v) + \epsilon$$

is equivalent to

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon, \\ \text{with} \quad x_1 &= v \\ x_2 &= v^2 \\ x_3 &= \cos(v) \\ x_4 &= \text{arctan}(v), \end{aligned}$$

where $v$ = original input variable, $x_i$ transformed input variables (features).

# Example

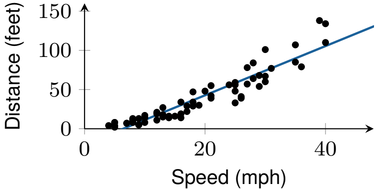
- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 5 \\ 1 & 7 \\ 1 & 7 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 39 \\ 1 & 39 \\ 1 & 40 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$

The normal equations $\Rightarrow \widehat{\beta} = \begin{bmatrix} -20.1 \\ 3.1 \end{bmatrix}$



# Example

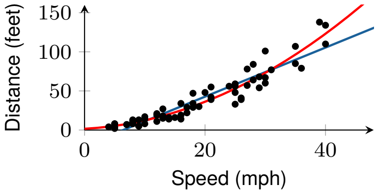
- $x$ = Speed
- $y$ = Distance

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

The normal equations $\Rightarrow \widehat{\beta} = \begin{bmatrix} 1.58 \\ 0.42 \\ 0.066 \end{bmatrix}$

$$\mathbf{X} = \begin{bmatrix} 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 5 & 25 \\ 1 & 5 & 25 \\ 1 & 5 & 25 \\ 1 & 7 & 49 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ \vdots & \vdots & \vdots \\ 1 & 39 & 1521 \\ 1 & 39 & 1521 \\ 1 & 40 & 1600 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 8 \\ 8 \\ 7 \\ 7 \\ 8 \\ \vdots \\ 138 \\ 110 \\ 134 \end{bmatrix}$$



UPPSALA UNIVERSITET

# Transforming the inputs

If the original input variable is $v$, we can use for instance:

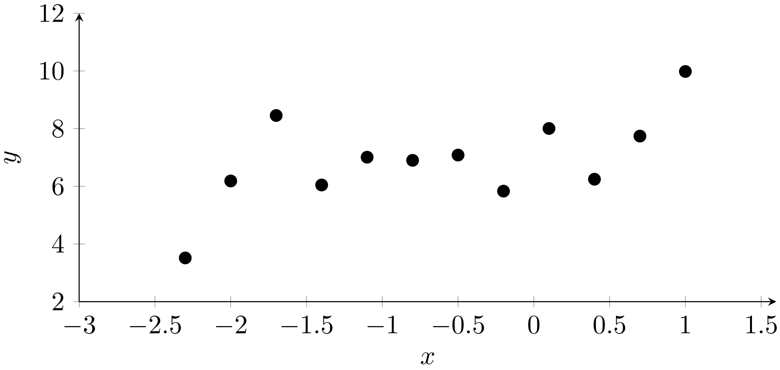
- a polynomial in $v$

$$y = \beta_0 + \beta_1 \underset{\substack{\| \\ x_1}}{v} + \beta_2 \underset{\substack{\| \\ x_2}}{v^2} + \beta_3 \underset{\substack{\| \\ x_3}}{v^3} + \cdots + \beta_p \underset{\substack{\| \\ x_p}}{v^p} + \epsilon$$

- radial basis function kernels (see book)
- ...

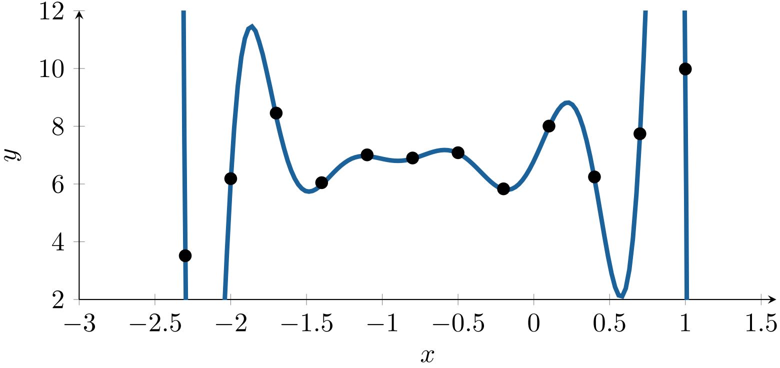
# Is the model too flexible?

With a $p = n - 1$ degree polynomial, we can fit $n$ data points perfectly.



# Is the model too flexible?

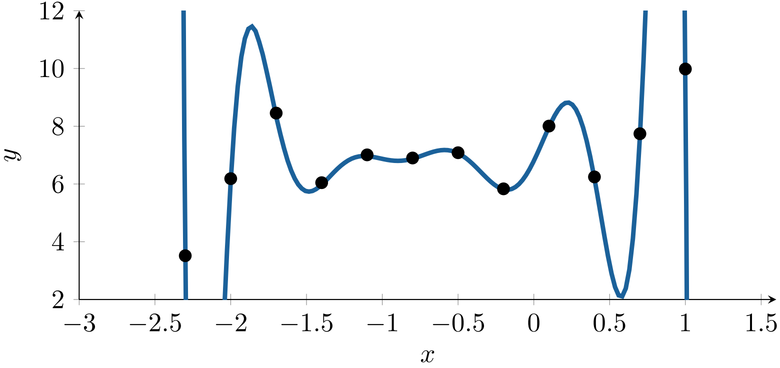
With a $p = n - 1$ degree polynomial, we can fit $n$ data points perfectly.




Is this desired?

# Is the model too flexible?

With a $p = n - 1$ degree polynomial, we can fit $n$ data points perfectly.




Is this desired? **Overfit!**

# Regularization

*“Keep $\beta$ small unless the data really convinces us otherwise”*

Least squares

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \, \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$$

# Regularization

*"Keep $\beta$ small unless the data really convinces us otherwise"*

Ridge regression = Least squares with $\ell_2$ regularizer

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\ \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_2^2$$

# Regularization

*"Keep $\beta$ small unless the data really convinces us otherwise"*

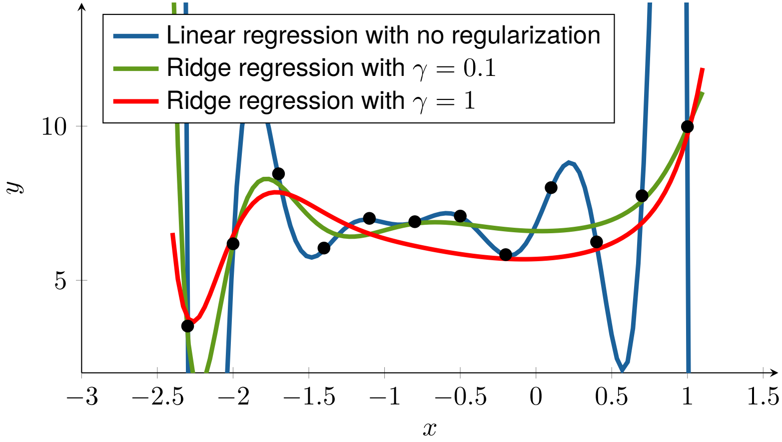
Ridge regression = Least squares with $\ell_2$ regularizer

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \; \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_2^2$$

$$\Rightarrow (\mathbf{X}^\mathsf{T}\mathbf{X} + \gamma\mathbf{I}_{p+1})\widehat{\boldsymbol{\beta}} = \mathbf{X}^\mathsf{T}\mathbf{y}$$

Here, $\gamma \geq 0$ is the regularization parameter.

# Is the model too flexible?




*Regularization can help us to avoid overfitting!*

# Regularization

## Ridge regression

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\ \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_2^2$$

(has a closed-form solution for $\widehat{\boldsymbol{\beta}}$)

## LASSO

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\ \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_1$$

(lacks a closed-form solution for $\widehat{\boldsymbol{\beta}}$)

Regularization can be used in many methods, not only linear regression!

UPPSALA UNIVERSITET

# Dummy variables for categorical inputs

For a categorical input with $2$ different classes/levels/labels A and B:
Create a dummy variable

$$x = \begin{cases} 0 & \text{if A} \\ 1 & \text{if B} \end{cases}$$

$$\Rightarrow y = \beta_0 + \beta_1 x + \varepsilon = \begin{cases} \beta_0 + \varepsilon & \text{if A} \\ \beta_0 + \beta_1 + \varepsilon & \text{if B} \end{cases}$$

# Dummy variables for categorical inputs

For a categorical input with $k = 4$ different classes/levels/labels A, B, C, D:
Create $k - 1 = 3$ dummy variables

$$x_1 = \begin{cases} 1 & \text{if B} \\ 0 & \text{if not B} \end{cases}, \quad x_2 = \begin{cases} 1 & \text{if C} \\ 0 & \text{if not C} \end{cases}, \quad x_3 = \begin{cases} 1 & \text{if D} \\ 0 & \text{if not D} \end{cases}$$

$$\Rightarrow y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon = \begin{cases} \beta_0 + \varepsilon & \text{if A} \\ \beta_0 + \beta_1 + \varepsilon & \text{if B} \\ \beta_0 + \beta_2 + \varepsilon & \text{if C} \\ \beta_0 + \beta_3 + \varepsilon & \text{if D} \end{cases}$$

# A few concepts to summarize lecture 2

**Regression** is about learning a model that describes the relationship between an input variable $\mathbf{x}$ (both numerical and categorical) and a numerical output variable $y$.

**Linear regression** corresponds to regression with a linear model.

**Maximum likelihood** with a Gaussian iid assumption on $\epsilon$
$\Rightarrow$ **least squares** and **normal equations**.

**Nonlinear transformations** can be applied to the input variables.

**Overfitting** is when the model adapts (too much) to noise in the data.

**Regularization** can help against overfitting.

**Categorical variables** are handled by dummy variables.