UPPSALA UNIVERSITET

Statistical Machine Learning

Lecture 2 – Linear regression, regularization

UPPSALA UNIVERSITET

David Sumpter
Division of Systems and Control
Department of Information Technology
Uppsala University.

Summary of Lecture 1 (I/III)

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**

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

UPPSALA UNIVERSITET

Summary of Lecture 1 (II/III)

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/III)

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** Estiimating paramaters. Normal distribution important

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

UPPSALA UNIVERSITET

Where are we now?

UPPSALA UNIVERSITET

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

We have developed this course over the last 4 years.

A book based on these notes will soon be published as a textbook with Cambridge University Press.

Please read in parallel to your studies.

Book website:

smlbook.org

Help us by reporting errors via practical GitHub interface:

github.com/uu-sml/sml-book-page/issues

Regression

- **Input variable** X
- **Output variable** Y

Regression: learning a model explaining Y from X, when Y is numerical.

$$Y = f(X; \beta) + \epsilon$$

β 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

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	
Adenine, Thymine, Cytosine, Guanine	
1 bike, 2 bikes, 5 bikes	

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	
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

Numerical or categorical variables

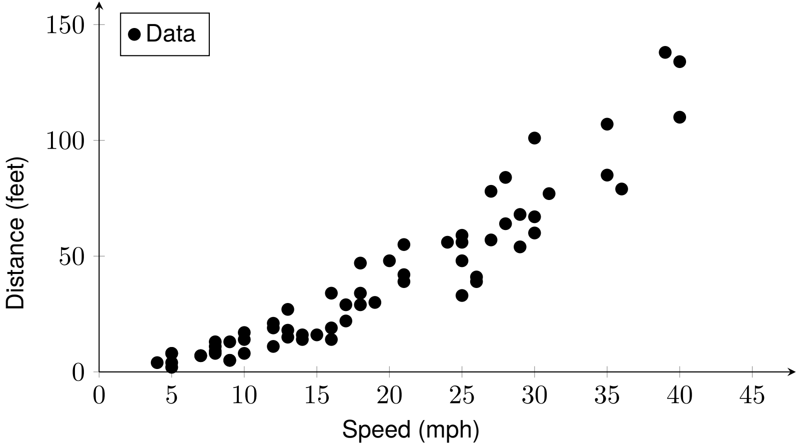
Numerical or categorical?

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

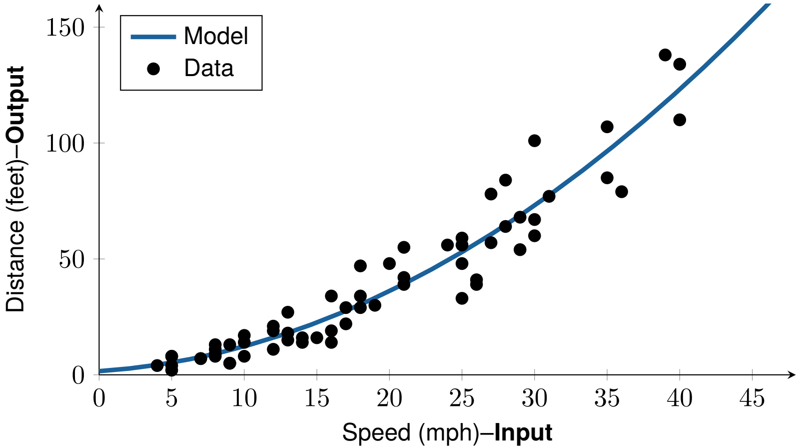
Categorical output variable? $\rightarrow$ classification. (But can be done with regression!)

Categorical input variable? Still regression!

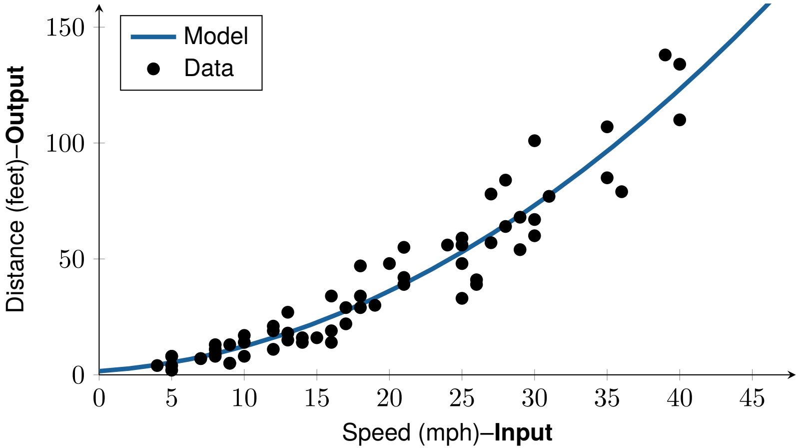
Regression example: car stopping distances



Regression example: car stopping distances



Regression example: car stopping distances



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

UPPSALA UNIVERSITET

Regression example: Alpha Go zero

Regression example: Alpha Go zero

- Input: State of the game (19×19 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.

Regression example: Alpha Go zero

- Input: State of the game (19×19 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",

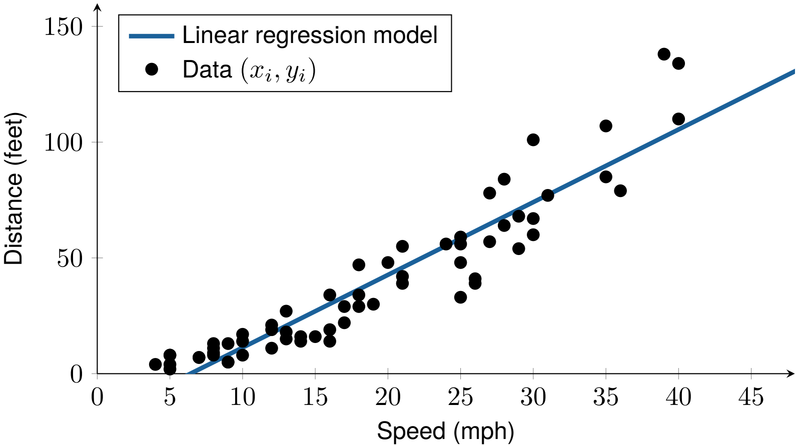
Output Y is linear combination of k inputs $X_1, \dots, X_k$ plus some noise/error ϵ,

$$Y = \underbrace{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k}_{f(X;\beta)} + \epsilon.$$

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

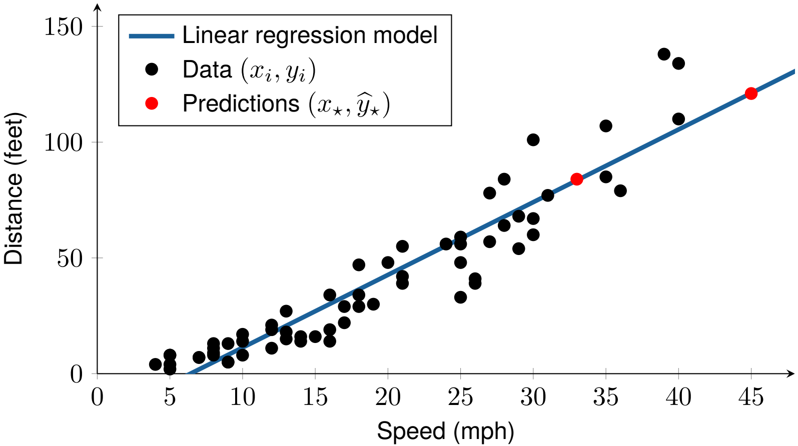
1. Learn/train/estimate model from training data $\mathcal{T}$: find $\widehat{\beta}_0, \widehat{\beta}_1, \dots, \widehat{\beta}_k$
2. Predict output for new test input using the model $\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \widehat{\beta}_2 X_2 + \cdots + \widehat{\beta}_k X_k$

Linear regression ($k = 1$)



UPPSALA UNIVERSITET

Linear regression ($k = 1$)



Learning the model from data

Linear regression model:

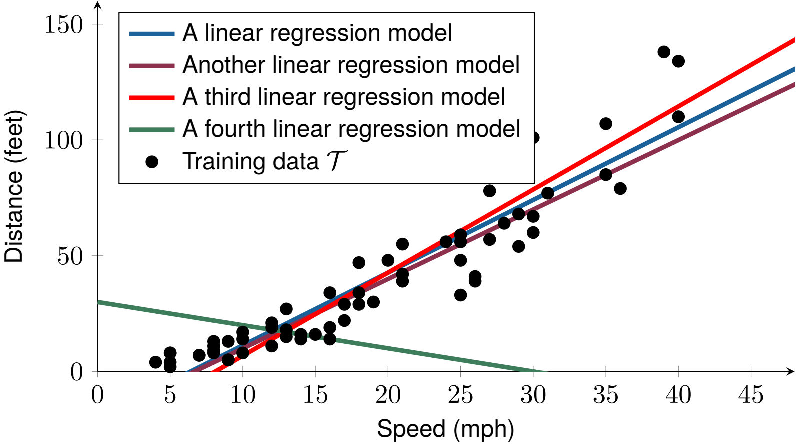
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

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

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

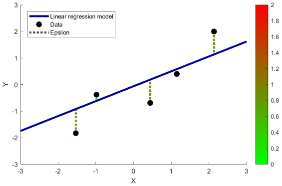
$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & -x_1^\mathsf{T}- \\ 1 & -x_2^\mathsf{T}- \\ \vdots & \vdots \\ 1 & -x_n^\mathsf{T}- \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

What is a good model?



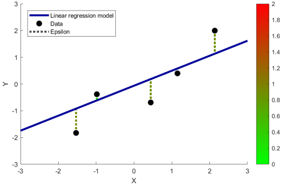
Learning using maximum likelihood

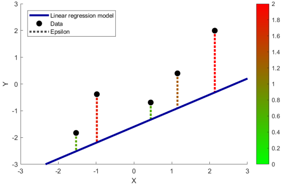
Learning a model from data is a matter of looking at the errors ε!

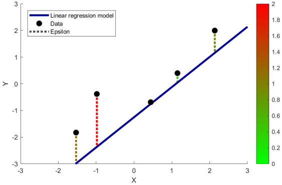


Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors ε!



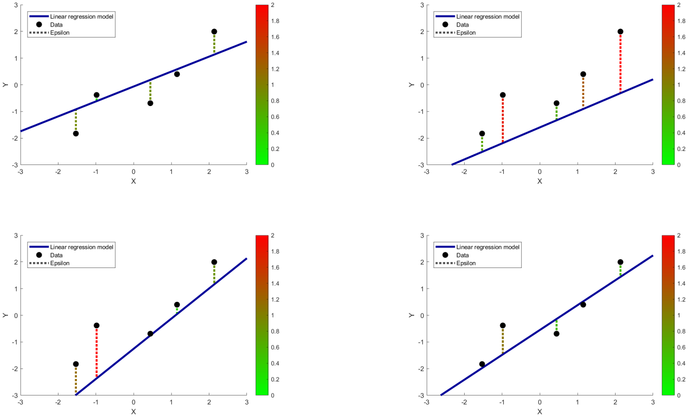




UPPSALA
UNIVERSITET

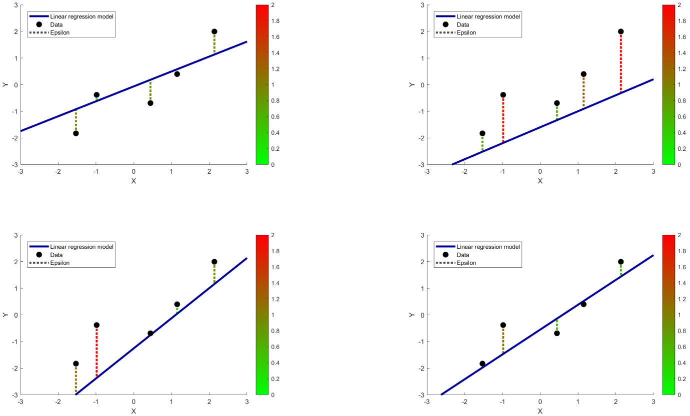
Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors ε!



Learning using maximum likelihood

Learning a model from data is a matter of looking at the errors ε!



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

Linear regression model in matrix form

Recall our linear regression model:

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

Assumptions (for now):

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

$$\Rightarrow P(y_i|x_i;\beta) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ip} - y_i)^2\right)$$

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

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

the **least squares** problem is achieved.

Least squares in matrix form

The least squares problem

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

Least squares in matrix form

The least squares problem

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

Minimize by differentiating and setting

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

Therefore,

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

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 (from lecture 1) $\mathbf{X}^\mathsf{T}\mathbf{X}$ 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 $k = 1$:

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

Linear regression: the key concepts

The linear regression model

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

+

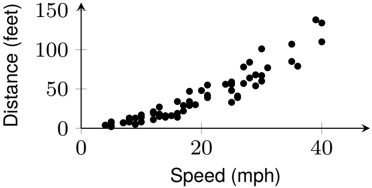
Maximum likelihood

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

$$\left.\begin{array}{c} \\ \\ \\ \end{array}\right\}$$ **Our first learning tool**

Example

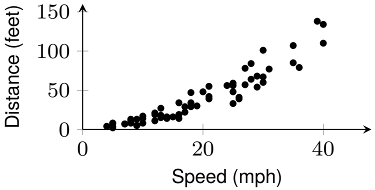
- x = Speed
- y = Distance



Example

- x = Speed
- y = Distance

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



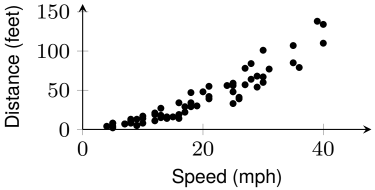
Example

- x = Speed
- y = Distance

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

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

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



Example

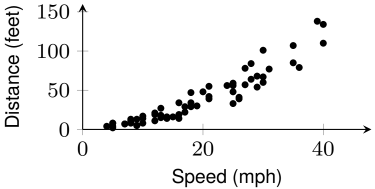
- x = Speed
- y = Distance

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

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

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

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



Example

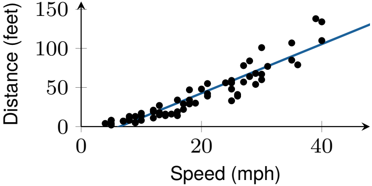
- x = Speed
- y = Distance

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

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

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

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



Example

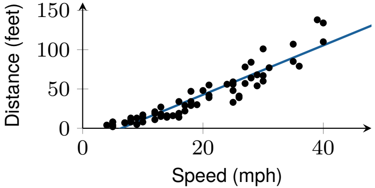
- x = Speed
- y = Distance

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

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

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

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

We can make arbitrary nonlinear transformations to the input variables!

The model is still a linear regression model, since

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

is equivalent to

$$\begin{aligned} y = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon, \\ \text{with} \quad & X_1 = v \\ & X_2 = v^2 \\ & X_3 = \cos(v) \\ & X_4 = \text{arctan}(v) \end{aligned}$$

x = original input variable, x_i transformed input variables (features).

Example

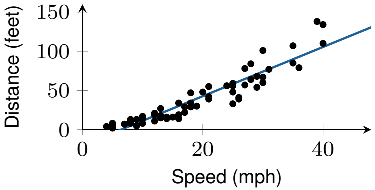
- x = Speed
- y = Distance

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

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

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

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



Example

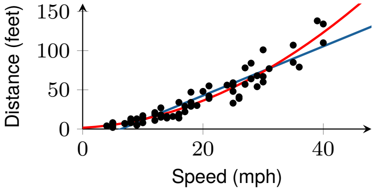
- x = Speed
- y = Distance

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

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

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

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



Transforming the inputs

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

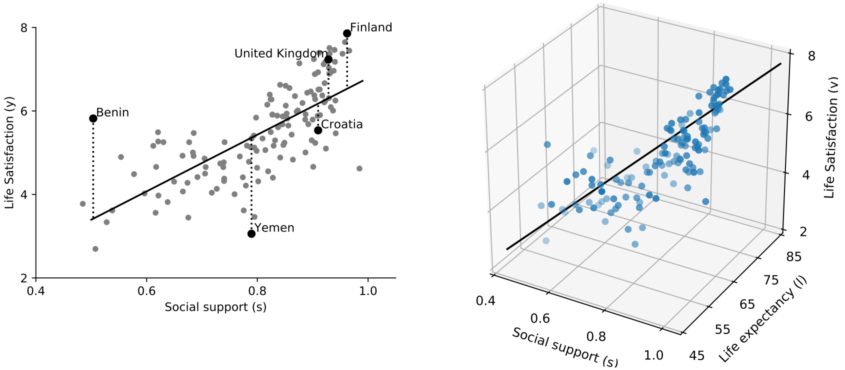
- a polynomial in v

$$y = \beta_0 + \beta_1 \underset{\substack{\| \\ x_1}}{v} + \beta_2 \underset{\substack{\| \\ x_2}}{v^2} + \beta_3 \underset{\substack{\| \\ x_3}}{v^3} + \cdots + \beta_k \underset{\substack{\| \\ x_k}}{v^k} + \varepsilon$$

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

Ex) Happiness

Happiness is fitted as a function of Log GDP, Social Support, Life Expectency, Freedom, Generosity and Corruption.



https://worldhappiness.report/ed/2019/

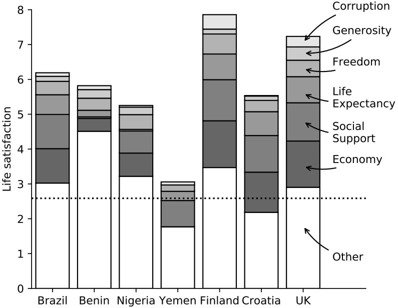
UPPSALA UNIVERSITET

Ex) Happiness

Happiness is fitted as a function of Log GDP, Social Support, Life Expectency, Freedom, Generosity and Corruption.

Table 2.1: Regressions to Explain Average Happiness across Countries (Pooled OLS)

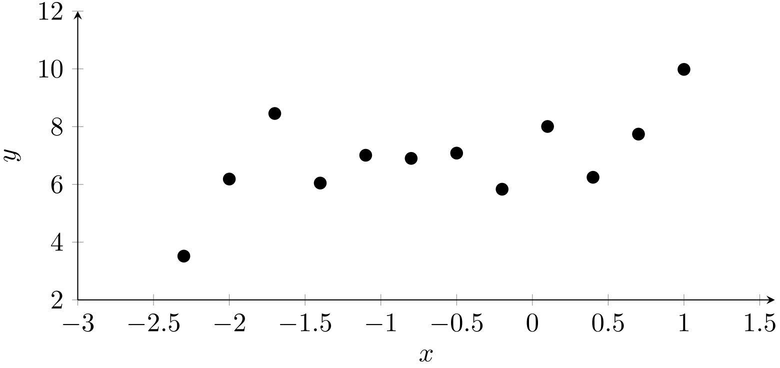
	Dependent Variable			
Independent Variable	Cantril Ladder (0-10)	Positive Affect (0-1)	Negative Affect (0-1)	Cantril Ladder (0-10)
Log GDP per capita	0.318	-.011	0.008	0.338
	(0.066)***	(0.01)	(0.008)	(0.065)***
Social support	2.422	0.253	-.313	1.977
	(0.381)***	(0.05)***	(0.051)***	(0.397)***
Healthy life expectancy at birth	0.033	0.001	0.002	0.03
	(0.01)***	(0.001)	(0.001)	(0.01)***
Freedom to make life choices	1.164	0.352	-.072	0.461
	(0.3)***	(0.04)***	(0.041)*	(0.287)
Generosity	0.635	0.137	0.008	0.351
	(0.277)**	(0.03)***	(0.028)	(0.279)
Perceptions of corruption	-.540	0.025	0.094	-.612
	(0.294)*	(0.027)	(0.024)***	(0.287)**
Positive affect				2.063
				(0.384)***
Negative affect				0.242
				(0.429)
Year fixed effects	Included	Included	Included	Included
Number of countries	157	157	157	157
Number of obs.	1,516	1,513	1,515	1,512
Adjusted R-squared	0.74	0.476	0.27	0.76



https://worldhappiness.report/ed/2019/

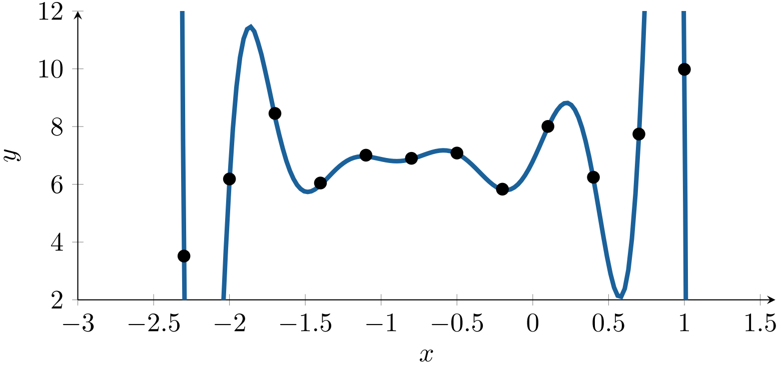
Is the model too flexible?

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



Is the model too flexible?

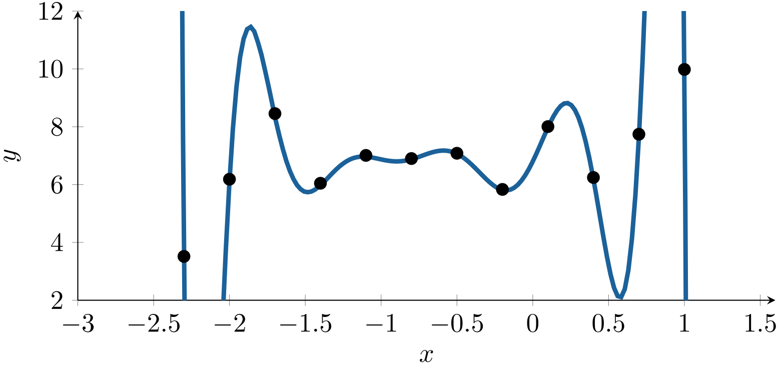
With a $k = n - 1$ degree polynomial, we can fit n data points perfectly.



Is this desired?

Is the model too flexible?

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



Is this desired? **Overfit!**

Regularization

"Keep β 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 β small unless the data really convinces us otherwise"

Least squares with Ridge regression

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

Regularization

"Keep $\boldsymbol{\beta}$ small unless the data really convinces us otherwise"

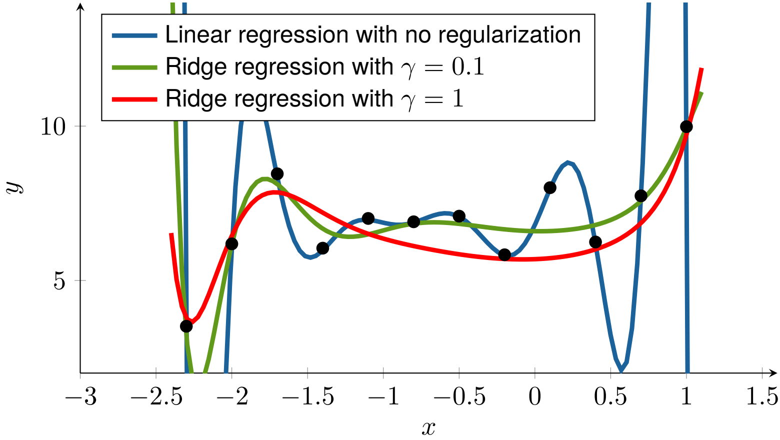
Least squares with Ridge regression

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

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

γ regularization parameter

Is the model too flexible?



Regularization can help us to avoid overfitting!

UPPSALA UNIVERSITET

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!

Dummy variables for categorical inputs

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

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

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

UPPSALA UNIVERSITET

Dummy variables for categorical inputs

For a categorical input with $a = 4$ different classes/levels/labels A, B, C, D:
Create $a - 1 = 3$ 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 Gaussian iid assumption on ε
$\Rightarrow$ **least squares** and **normal equations**

Nonlinear transformations can be applied to the input variables

Overfit is when the model adapts (too much) to noise in the data

Regularization can help against overfitting

Categorical variables are handled by dummy variables