Linear Regression, SVMs

Jose Martinez Heras

08/03/2018

Resources

Watch the video of this lecture

https://dlmultimedia.esa.int/download/public/videos/2048/03/004/4803_004_AR_EN.mp4

Watch the practical exercise video

https://dlmultimedia.esa.int/download/public/videos/2048/03/003/4803_003_AR_EN.mp4

Get presentation and additional resources on

https://github.com/jmartinezheras/2018-MachineLearning-Lectures-ESA

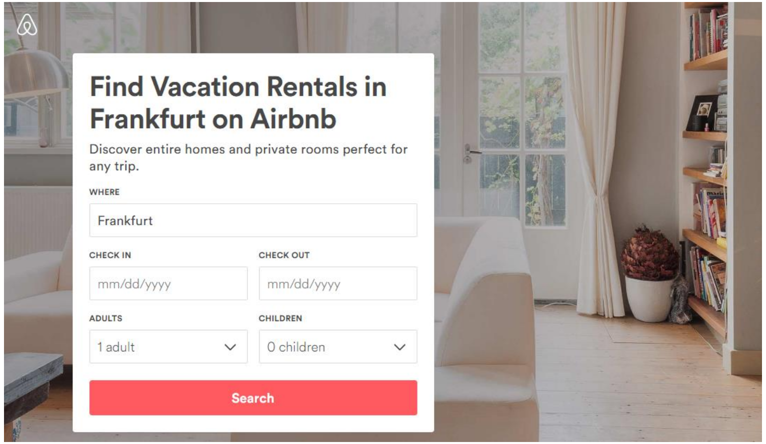
Outline for Supervised Learning (1)

Supervised Learning (1)

- Linear, polynomial regression
- Lasso, Ridge, ElasticNet regression
- Logistic Regression
- Support Vector Machines (SVM)

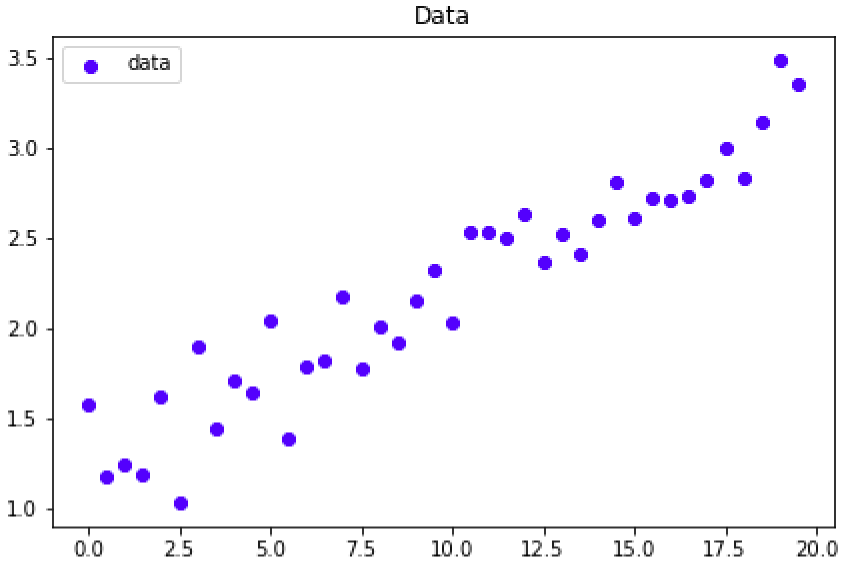
- Hands-on Supervised Learning

Predict price of vacation rentals in Frankfurt on Airbnb



Linear Regression

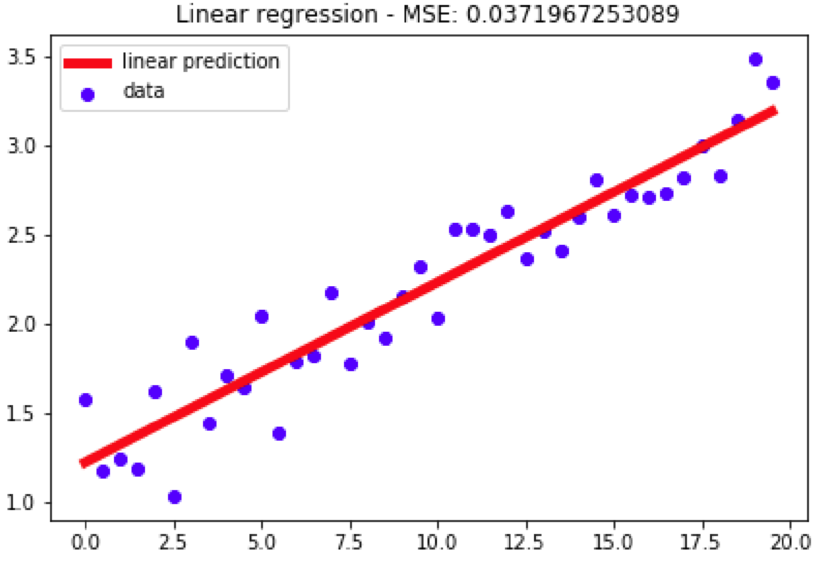
Let’s create some data



$$y = 0.1x + 1.25 + 0.2GaussianNoise$$

Linear Regression

Let's perform linear regression…



$$y = 0.1x + 1.25 + 0.2GaussianNoise$$

$$y = wx + b$$

w = 0.1014
b = 1.2258

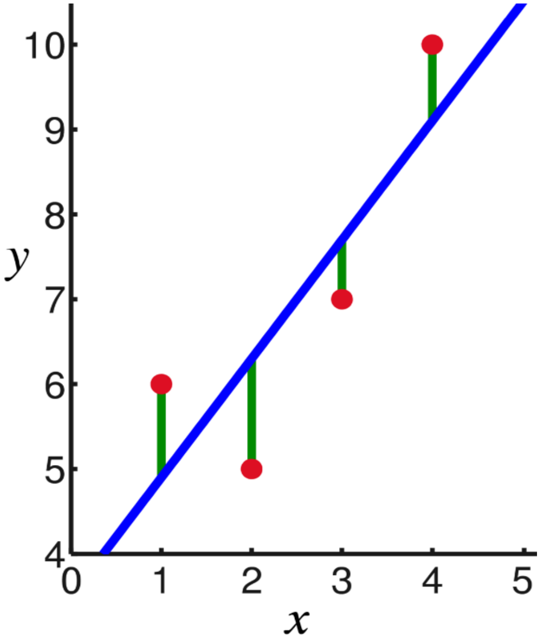
$$y = 0.1014x + 1.2258$$

How we fitted the line?

We just found the values of 'w' and 'b' that minimize the Mean Squared Error

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(Y_i - \widehat{Y}_i\right)^2$$

Mean Squared Error

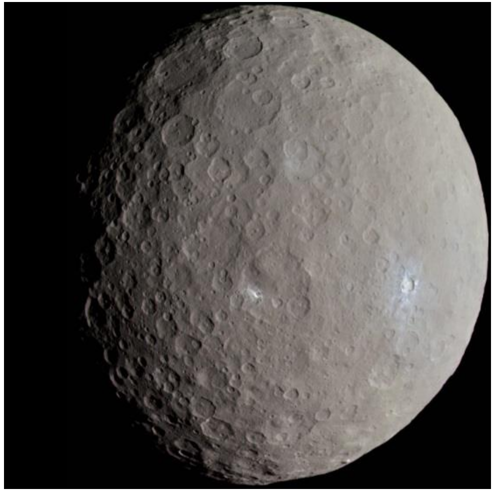


How we fitted the line?

How do we know which values of 'w' and 'b' minimize the Mean Squared Error?

Least squares method

Carl Friedrich Gauss



Ceres asteroid

Picture by Justin Cowart - Ceres - RC3 - Haulani Crater, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=49700320

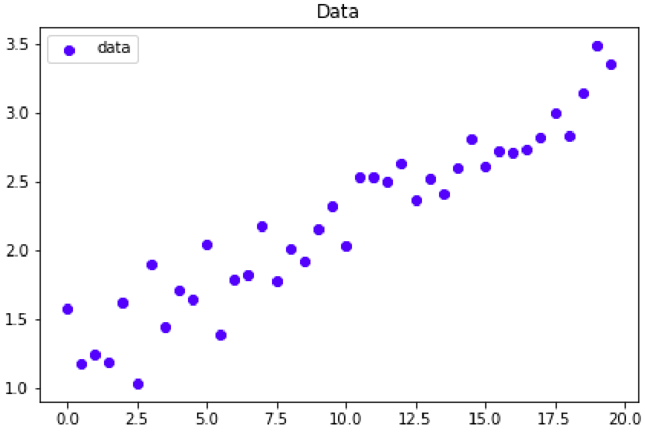
Least Squares - notation

$$y = wx + b = b + wx$$

$$X = 1, x$$

$$W = b, w$$

$$y = WX = b + wx$$



Also called features

$$\boxed{X} = \boxed{x_0, x_1, x_2, x_3, \dots, x_n} \qquad x_0 = 1$$

$$W = w_0, w_1, w_2, w_3, \dots, w_n \qquad w_0 = b$$

$$y = WX = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

x_0	x_1	y
1	0.0	1.57
1	0.5	1.18
1	1.0	1.24
1	1.5	1.19
1	2.0	1.62

X = matrix (m,n)
Y = vector (m)

m = number of samples
n = number of features

How we fitted the line?

Which values of line parameters minimize the Mean Squared Error?

Least squares Method

Carl Friedrich Gauss

Least Squares Method

$$\boxed{\widehat{W} = (X^T X)^{-1} X^T y}$$

$$X = x_0, x_1, x_2, x_3, \dots, x_n \qquad x_0 = 1$$

$$W = w_0, w_1, w_2, w_3, \dots, w_n \qquad w_0 = b$$

$$y = WX = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

$\widehat{W}$ is the best approximation to W

How we fitted the line?

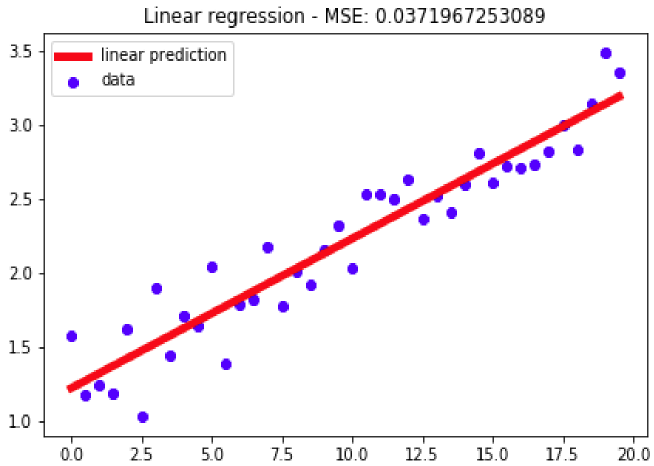
Which values of line parameters minimize the Mean Squared Error?

Least squares Method

Carl Friedrich Gauss

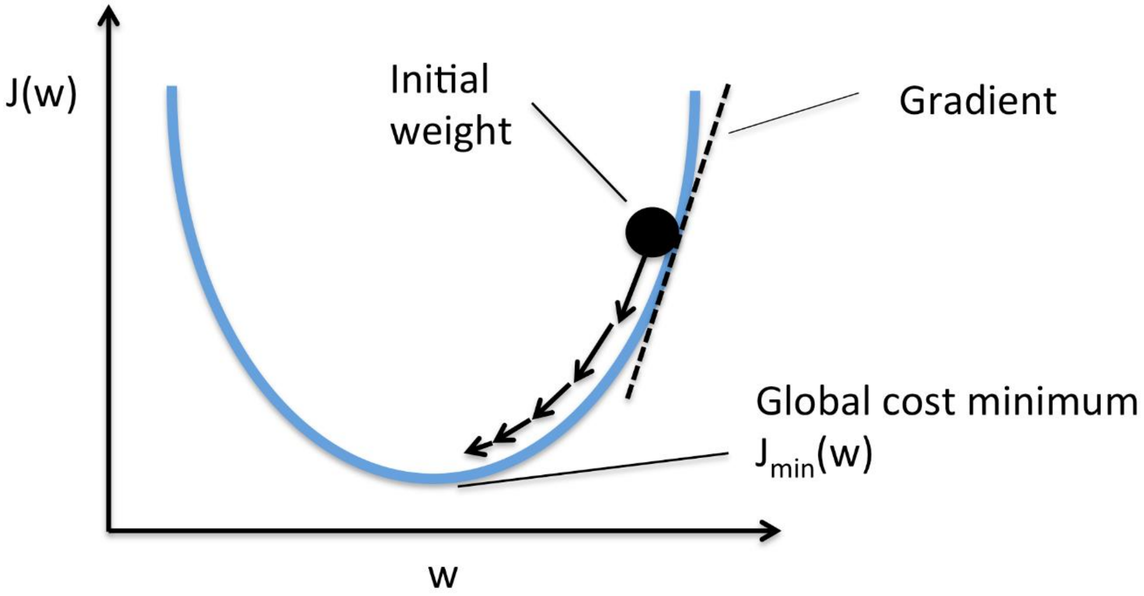
Least Squares Method

$$\widehat{W} = (X^T X)^{-1} X^T y$$



Gradient Descent

There is another way: Gradient Descent



Gradient Descent Visualization. Credit: rasbt.github.io

$$J = MSE = \frac{1}{m}\sum^{m}\left(Y_i - \widehat{Y}_i\right)^2$$

Learning

$$W = W - \alpha \frac{\partial J}{\partial W}$$

Gradient Descent

There is another way: Gradient Descent

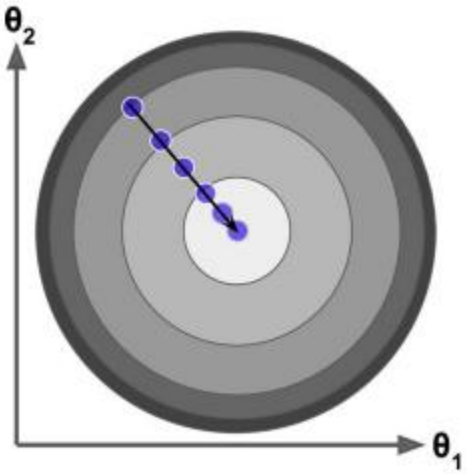


Image Credits: Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (pp. 113-114). O'Reilly Media. Kindle Edition.

Gradient Descent

When using Gradient Descent we need to **normalize** the inputs

- "normalize" means, put every input in a similar scale
- E.g. predict price of a property: n_reviews = [0 – 500], rooms = [1, 8]

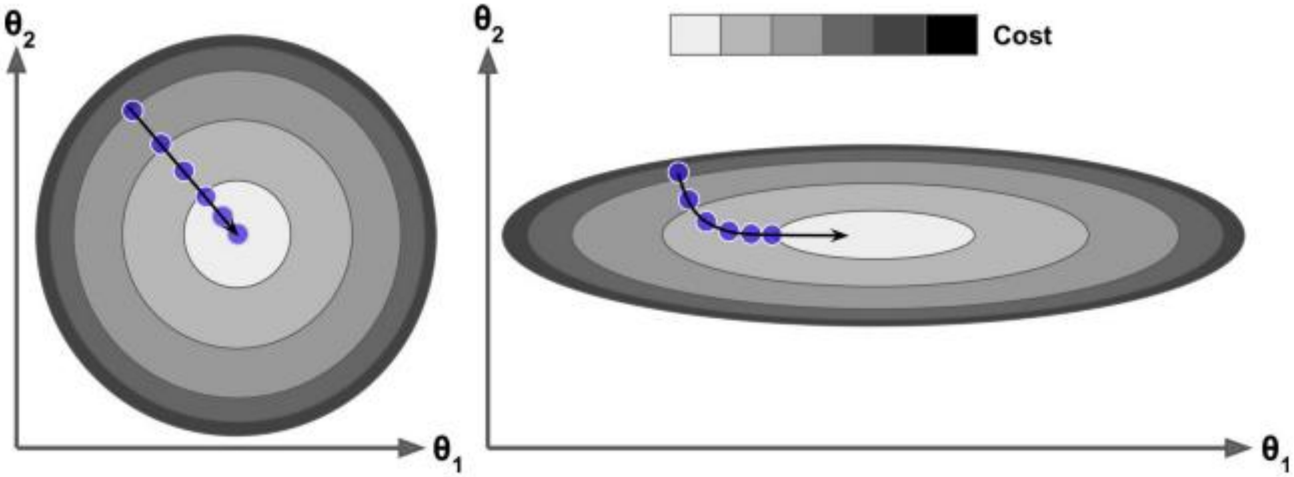
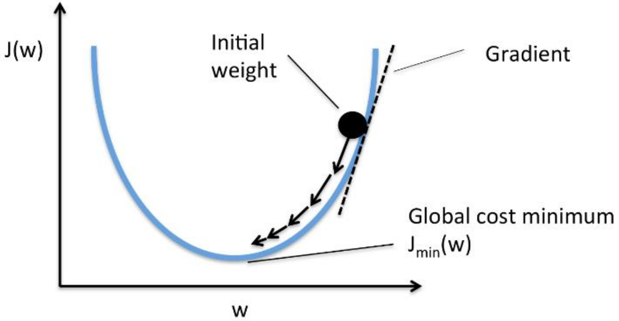


Image Credits: Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (pp. 113-114). O'Reilly Media. Kindle Edition.

When we use which method?

$$\hat{W} = (X^T X)^{-1} X^T y$$



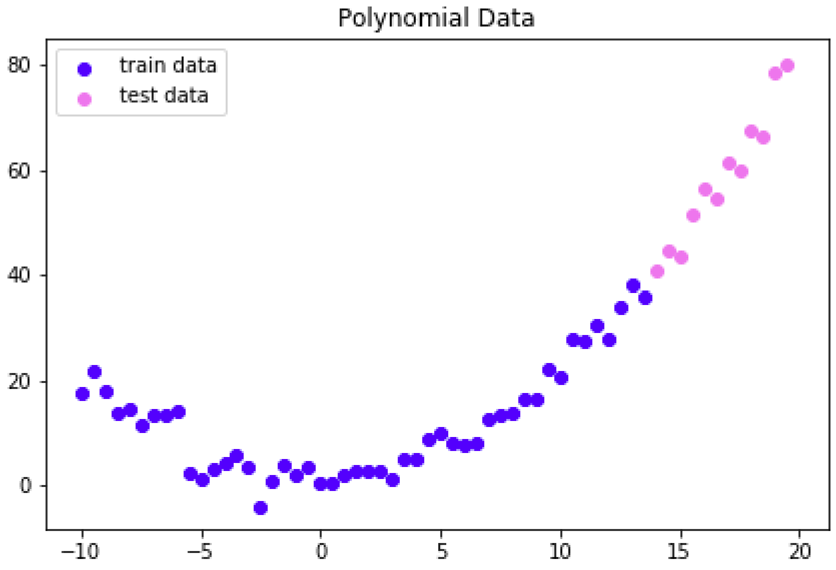
Least Squares

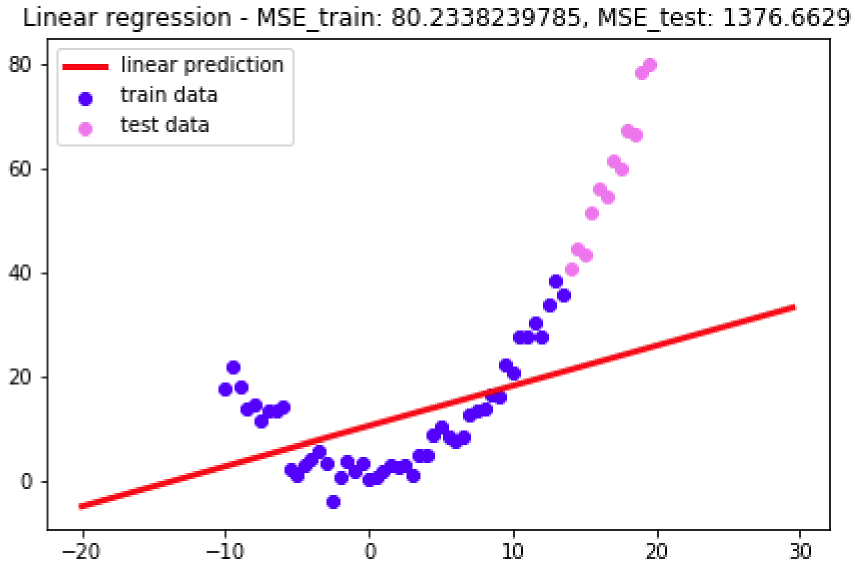
- when there is a relatively small number of features (< 1,000)

Gradient Descent

- when there are many features (> 1,000)
- when we need to stop training at any time
    - e.g. if we only have 1 minute
- If data does not fit in memory
- If you have new data (e.g. stream) and don't want to start all over (with all previous data)

Polynomial Regression





$$y = 0.2x^2 + 0.1x + 1 + 3GaussianNoise$$

Linear Regression: $y = w_0 + w_1 x$

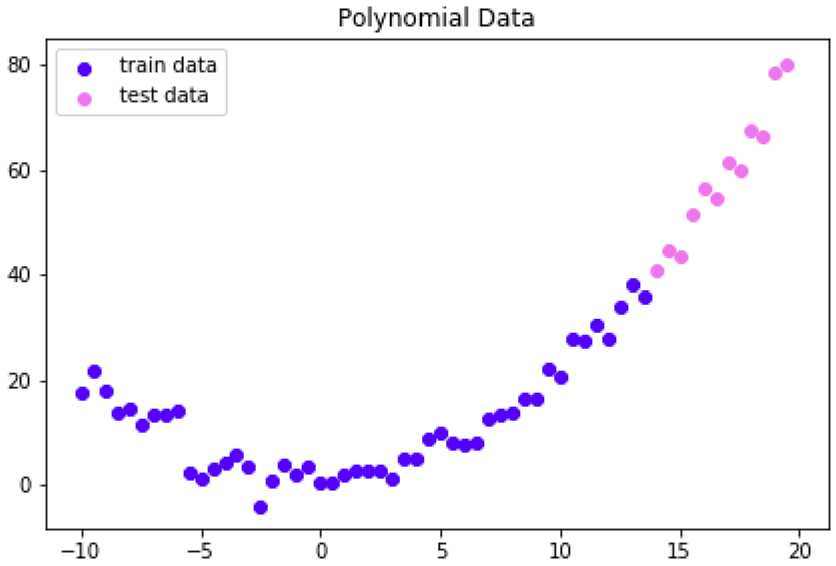
Polynomial Regression

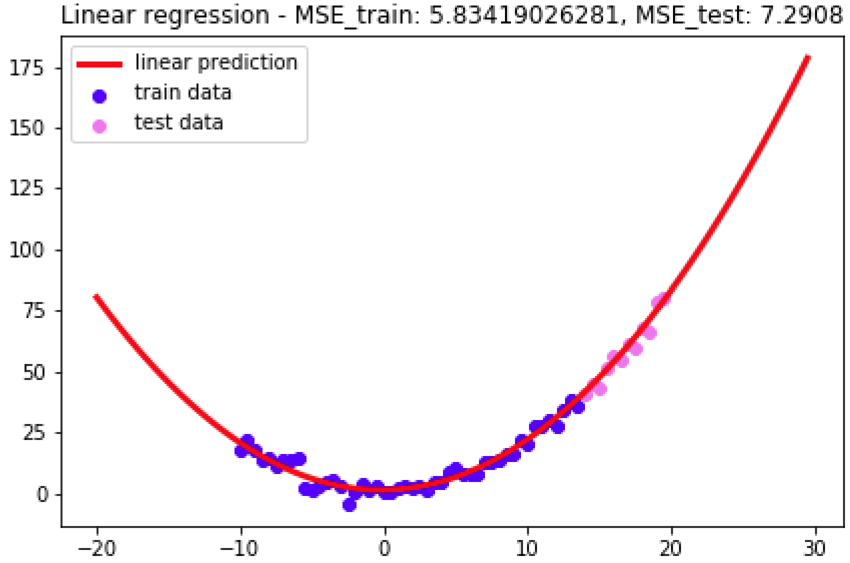
- You already know how to do it
- It is not a new technique, it's a **feature**

	x	
x_0	x_1	y
1	-10.0	17.74
1	-9.5	21.86
1	-9	17.84
1	-8.5	13.71
1	-8	14.47

	x	x^2	
x_0	x_1	x_2	y
1	-10.0	100.0	17.74
1	-9.5	90.25	21.86
1	-9	81.00	17.84
1	-8.5	72.25	13.71
1	-8	64.00	14.47

Polynomial Regression





$y = 0.2x^2 + 0.1x + 1 + 3GaussianNoise$

Linear Regression: $y = w_0 + w_1 x + w_2 x^2$

Polynomial Regression

- Polynomial Regression = Linear Regression with polynomial features
- You can get creative:
  - x^2, x^3, x^4…
  - zx^2, zx^3, z^2x^2, …

	x	x^2	
x_0	x_1	x_2	y
1	-10.0	100.0	17.74
1	-9.5	90.25	21.86
1	-9	81.00	17.84
1	-8.5	72.25	13.71
1	-8	64.00	14.47

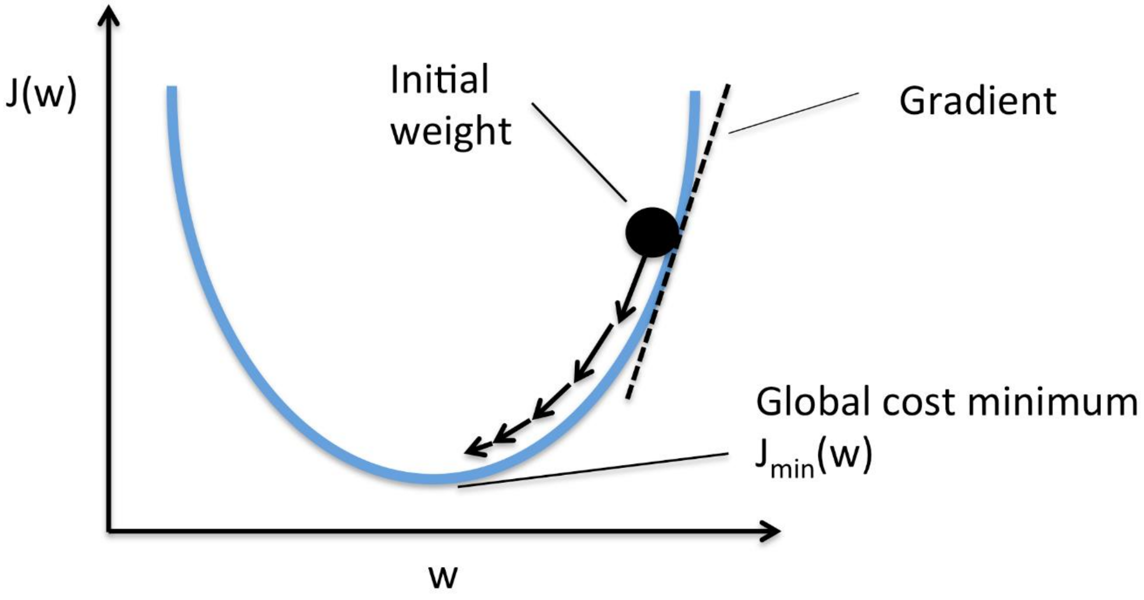
Features: x_0, x_1, x_2

What if some of the inputs are irrelevant?

- Ridge Regression
- Lasso Regression
- ElasticNet Regression

Ridge Regression

Remember Gradient Descent?



Gradient Descent Visualization. Credit: rasbt.github.io

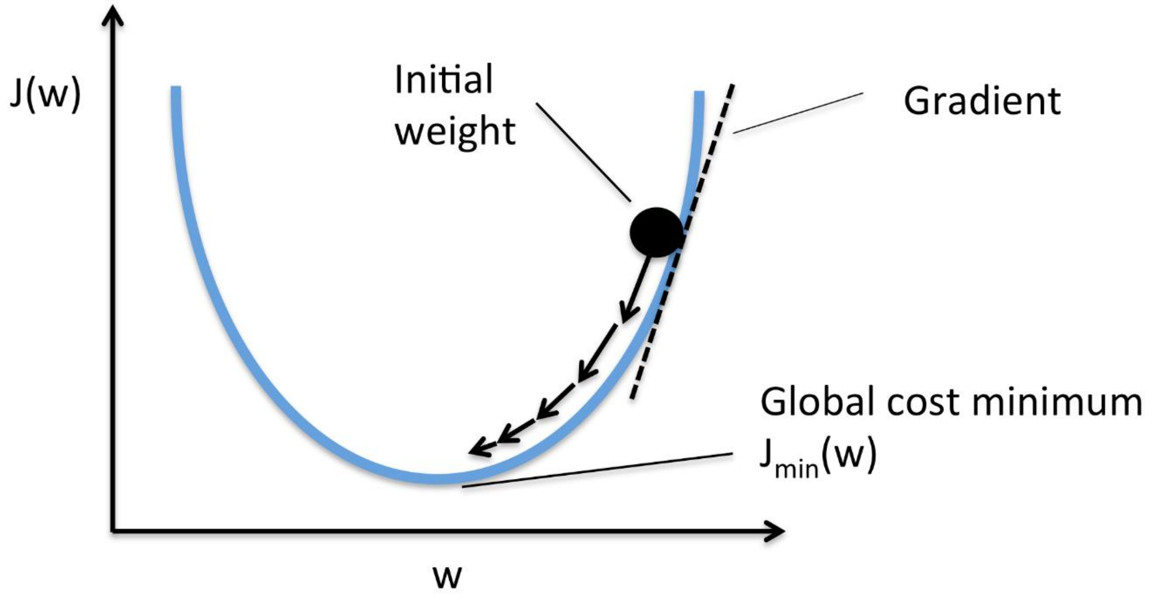
$$J = MSE = \frac{1}{m}\sum^{m}(Y_i - \widehat{Y}_i)^2$$

Learning

$$W = W - \alpha \frac{\partial J}{\partial W}$$

Ridge Regression

Upgrade the Cost Function with a **regularization** term



Gradient Descent Visualization. Credit: rasbt.github.io

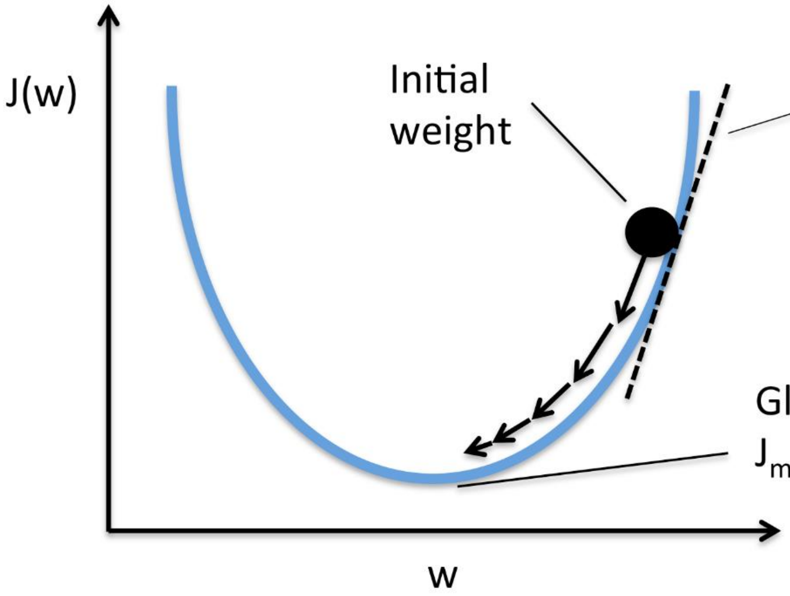
$$J = MSE$$

$$J = MSE + \boxed{\alpha \frac{1}{2} \sum_{j=1}^{n} w_j^2}$$

$l2\ penalty$

Lasso Regression

Upgrade the Cost Function with a **regularization** term



Gradient Descent Visualization. Credit: rasbt.github.io

$$J = MSE$$

$$J = MSE + \boxed{\alpha \sum_{j=1}^{n} |w_j|}$$

$l1\ penalty$

ElasticNet Regression

Regularization combining Ridge and Lasso regularizations

$$J = MSE + r \cdot Lasso + (1 - r) \cdot Ridge$$

$$J = \frac{1}{m}\sum^{m}(Y_i - \widehat{Y}_i)^2 + \boxed{r \cdot \alpha \sum_{j=1}^{n}|w_j| + \alpha \frac{1-r}{2}\sum_{j=1}^{n} w_j^2}$$

Which Linear Regression?

In general, it is always a good idea to use some regularization

Ridge

- few irrelevant features
- some correlated features

Lasso

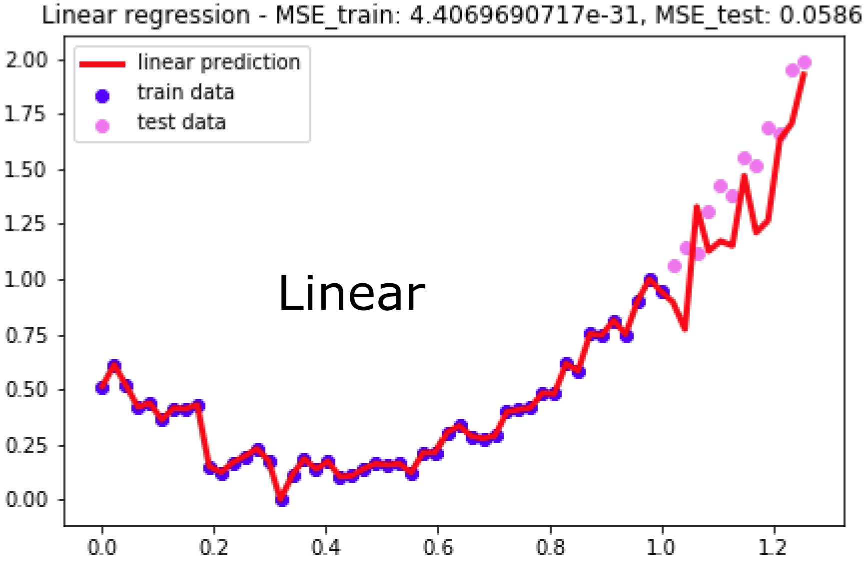
- many irrelevant features
- little correlation among features

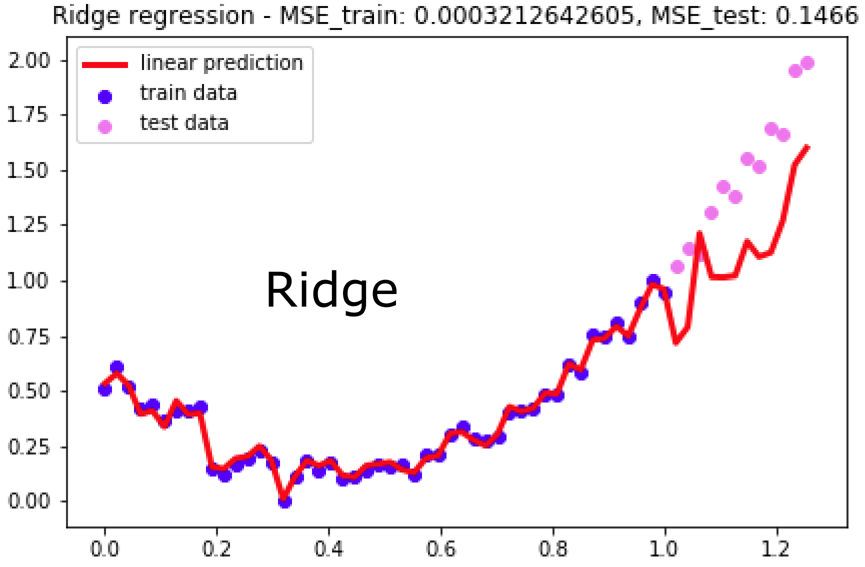
ElasticNet

- Large number of features
- Possibly many irrelevant
- Possibly correlated features

When to use which Linear Regression?

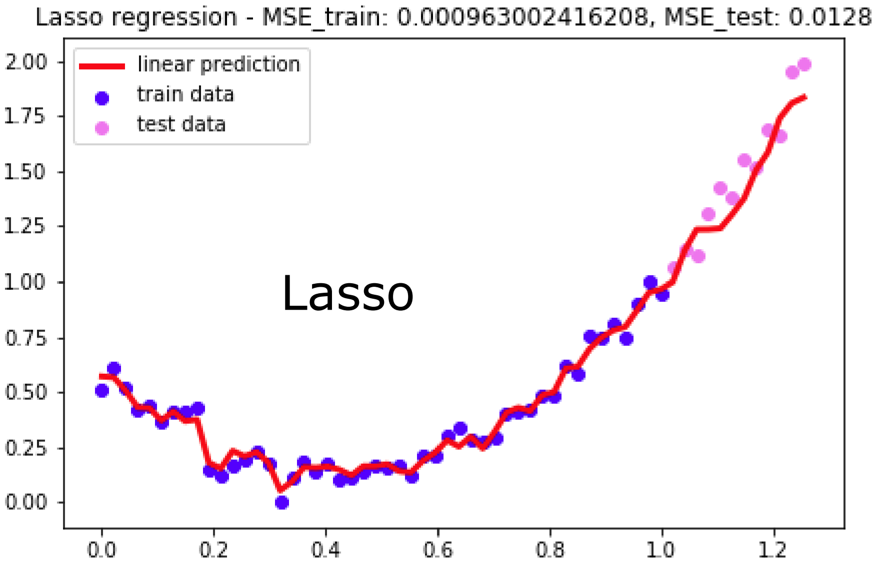
Let's add 50 irrelevant features (Gaussian Noise)

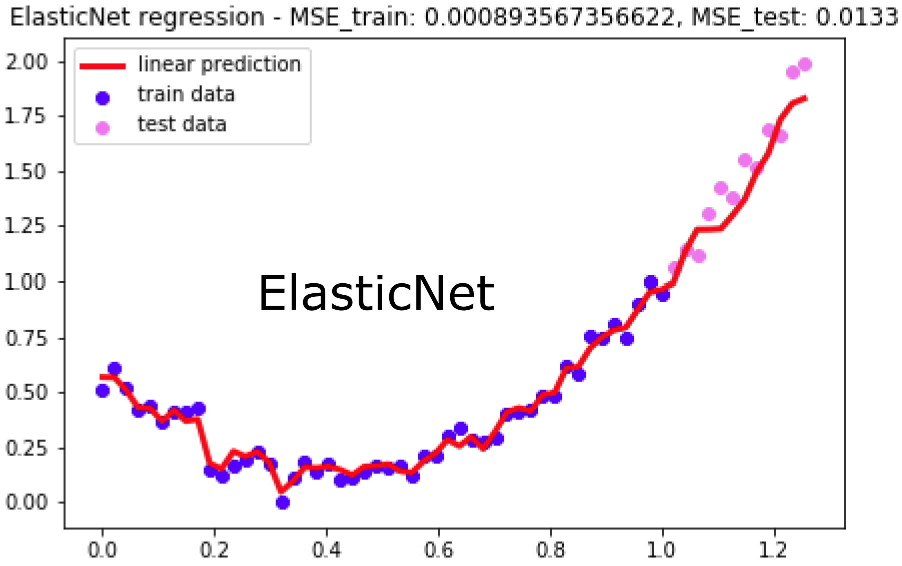




When to use which Linear Regression?

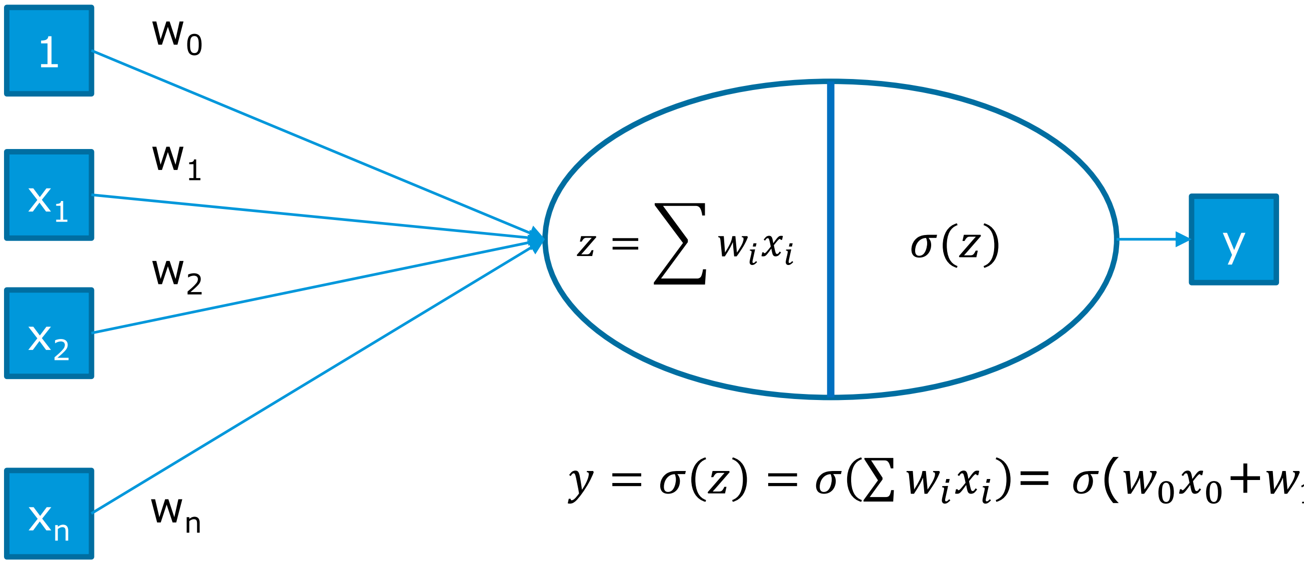
Let’s add 50 irrelevant features (Gaussian Noise)





Logistic Regression

- Tiny Neural Network used for classification
  - It has exactly 1 neuron

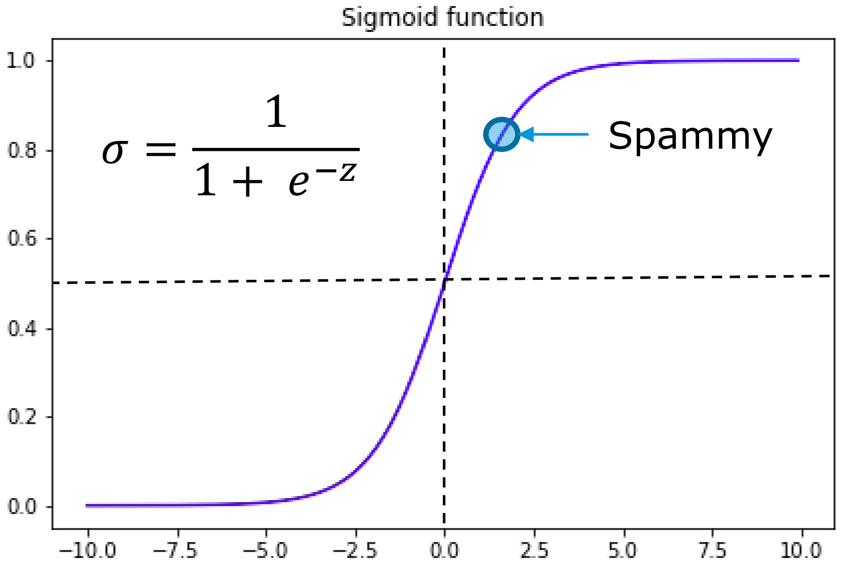


$$y = \sigma(z) = \sigma(\sum w_i x_i) = \sigma(w_0 x_0 + w_1 x_1 + \ldots w_n x_n)$$

Logistic Regression

Sigmoid Function

- Values [0, 1]
- Used for estimating probability
    - Spam = 1
    - Not spam = 0
- In binary classification:
    - 1 if $p \geq 0.5$
    - 0 if $p < 0.5$



Logistic Regression Example

Chance of passing an exam based on how much you studied

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

$$p = \frac{1}{1 + e^{(-(1.5046 \cdot hours - 4.0777))}}$$

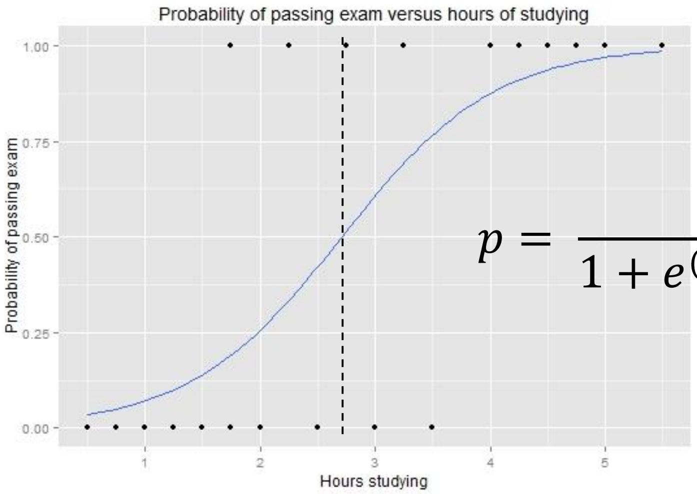
Hours	Probability of passing
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Wikipedia contributors, "Logistic regression," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=827666692 (accessed March 5, 2018).

Logistic Regression Example

Chance of passing an exam based on how much you studied

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



$$p = \frac{1}{1 + e^{(-(1.5046 \cdot hours - 4.0777))}}$$

Hours	Probability of passing
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Wikipedia contributors, "Logistic regression," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=827666692 (accessed March 5, 2018).

Logistic Regression

Cost: log loss

$$J = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)]$$

- No formula to solve it
- **Only numerical optimization** - Gradient Descent
- We can also add l1 or l2 **regularization** terms

What about if there are more than 2 classes?

Iris setosa

Iris versicolor

Iris virginica

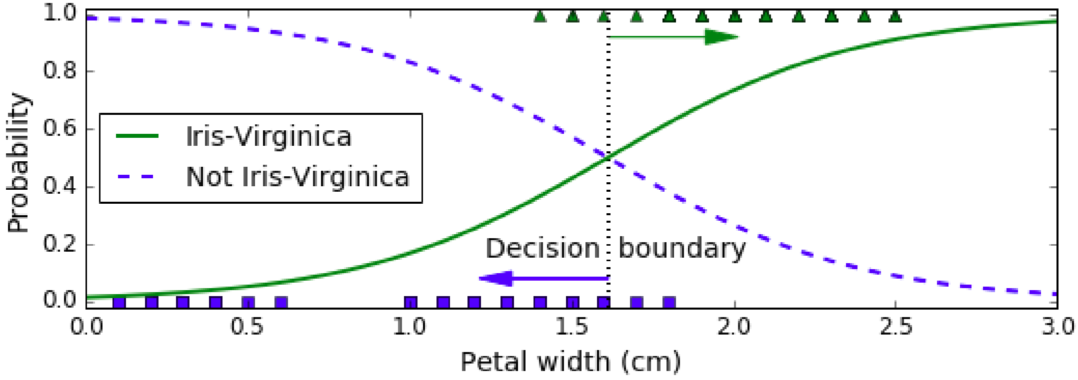
Knowing the sepal and petal length and width, which flower it is?

Pictures from Wikipedia contributors, "Iris flower data set," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Iris_flower_data_set&oldid=824486644 (accessed March 5, 2018).

What about if there are more than 2 classes?

Transform the problem into binary classification

- Setosa vs non-setosa
- Versicolor vs non-versicolor
- Virginica vs non-virginica



Machine Learning libraries can handle multiclass classification for us

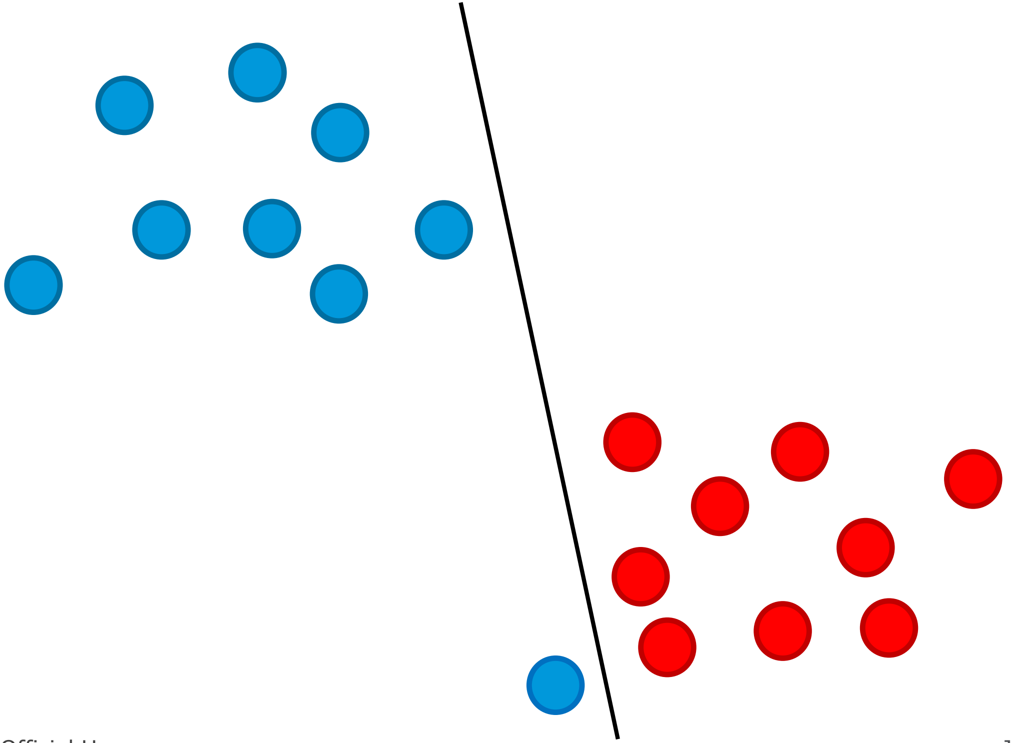
Visualization from https://github.com/ageron/handson-ml/blob/master/04_training_linear_models.ipynb

Support Vector Machines

What’s the optimal way to do classification?

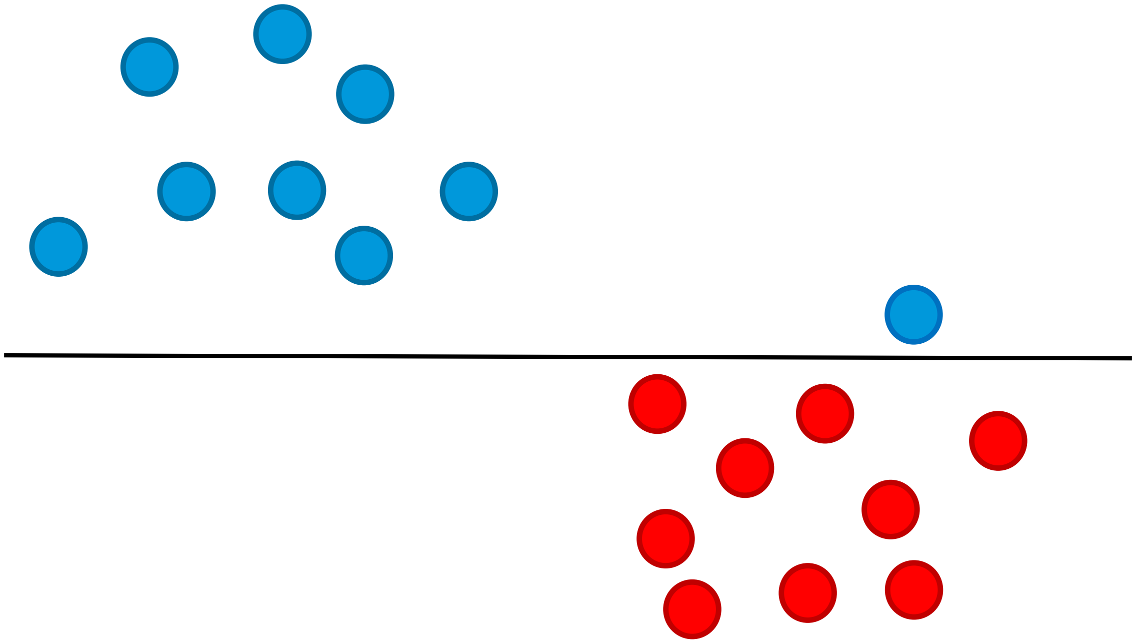
Support Vector Machines

What’s the optimal way to do classification?



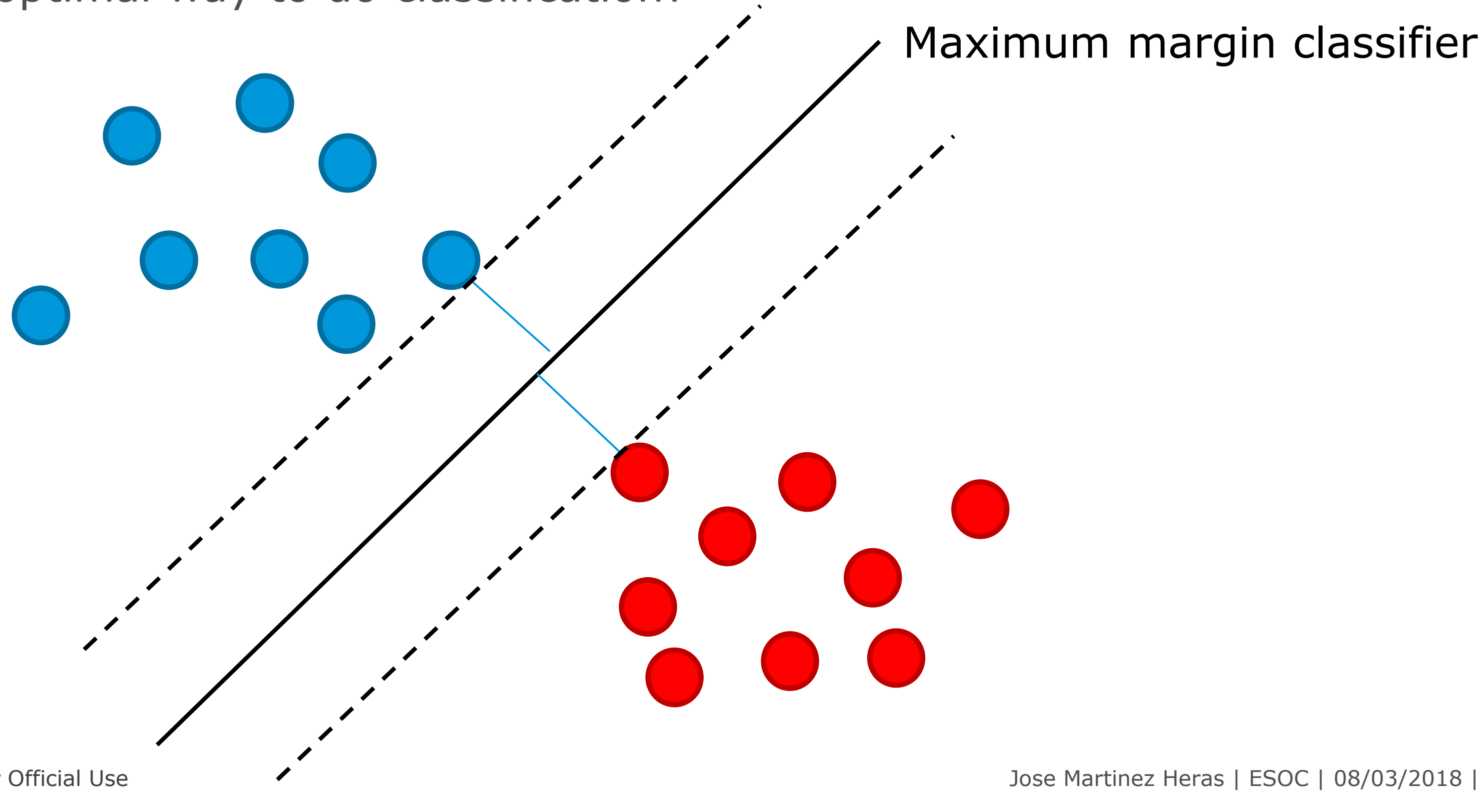
Support Vector Machines

What's the optimal way to do classification?



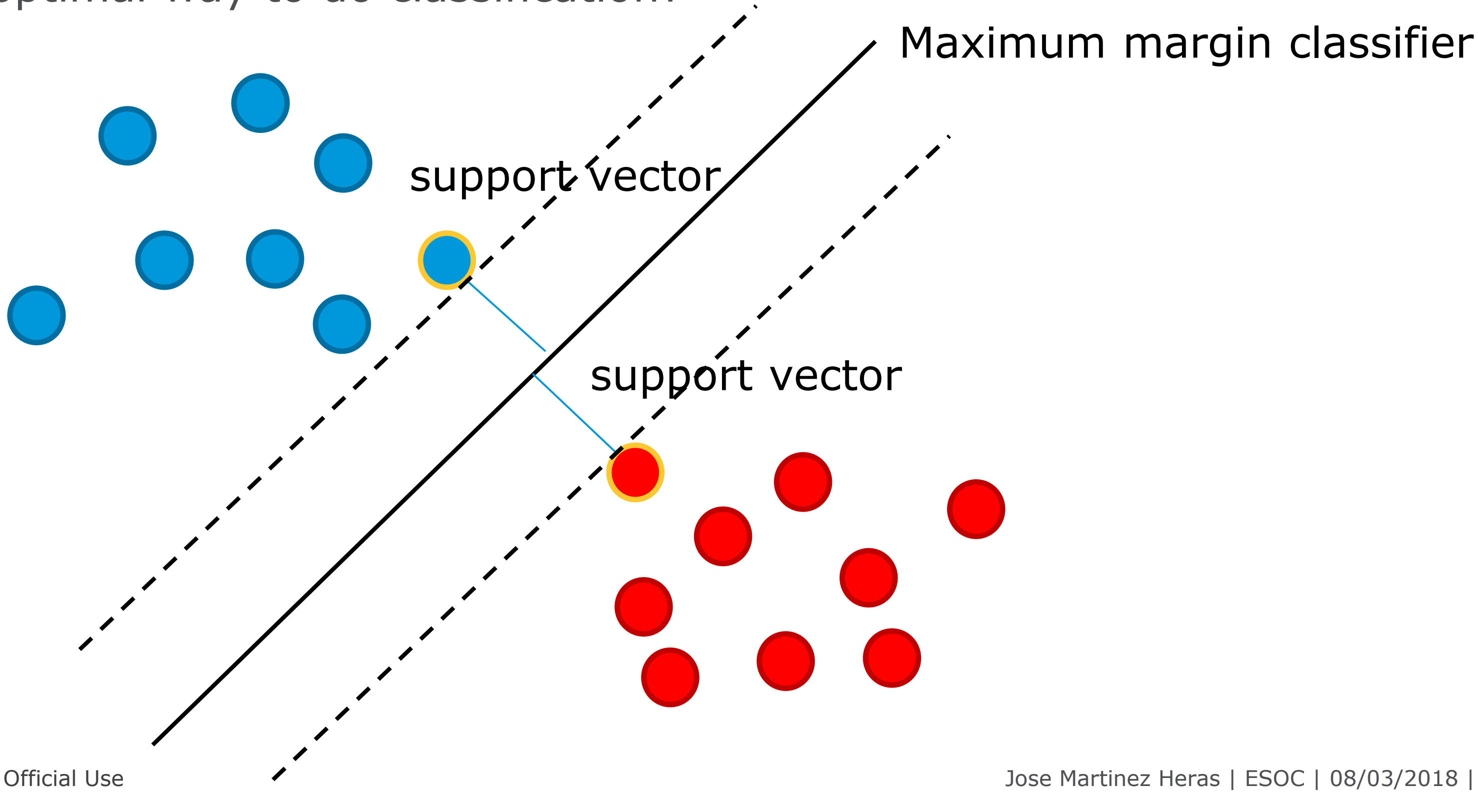
Support Vector Machines

What's the optimal way to do classification?



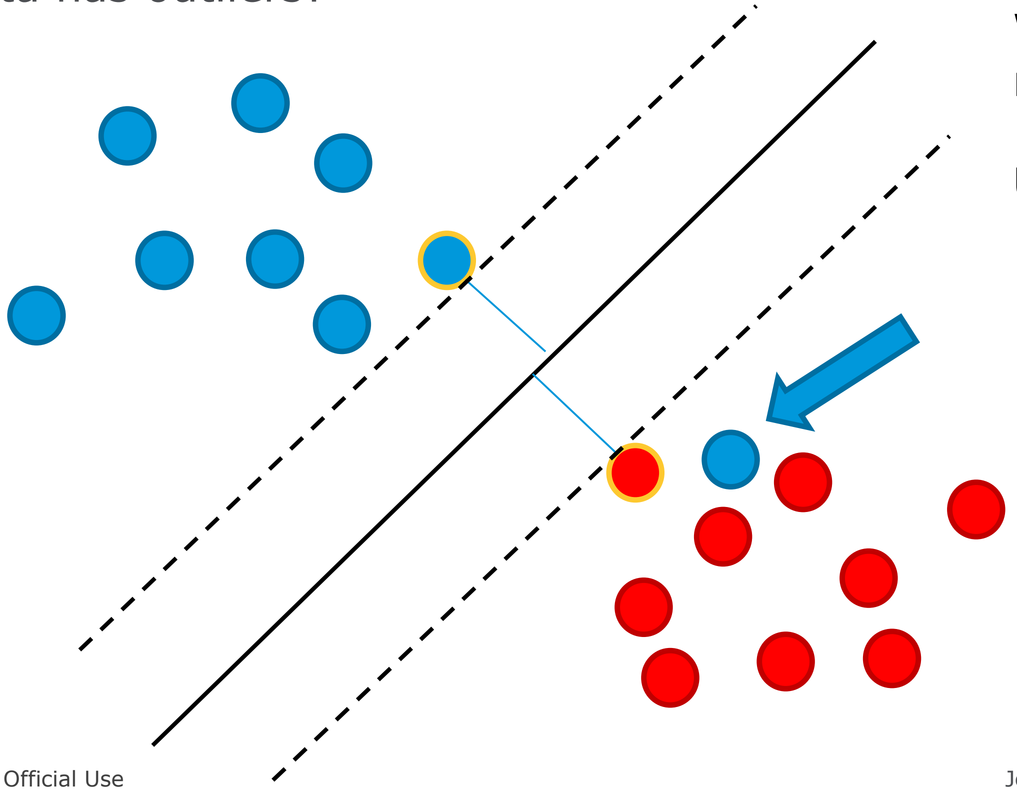
Support Vector Machines

What's the optimal way to do classification?



Support Vector Machines

What's if data has outliers?



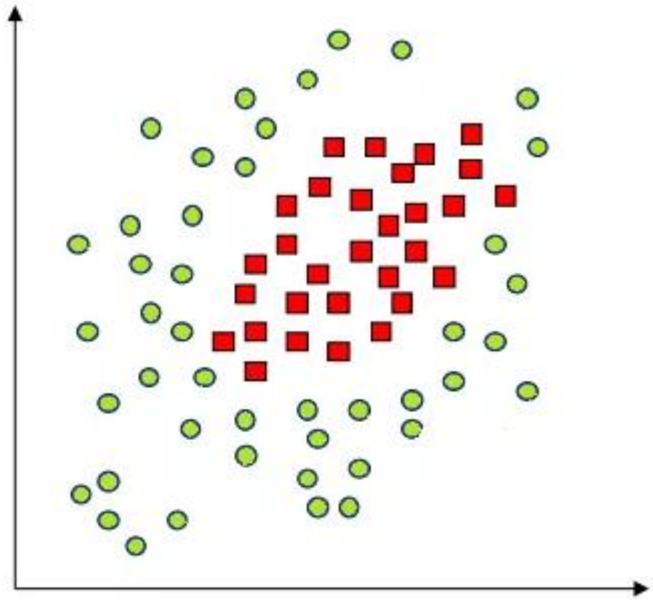
We still want a maximum margin

Use penalty parameter C

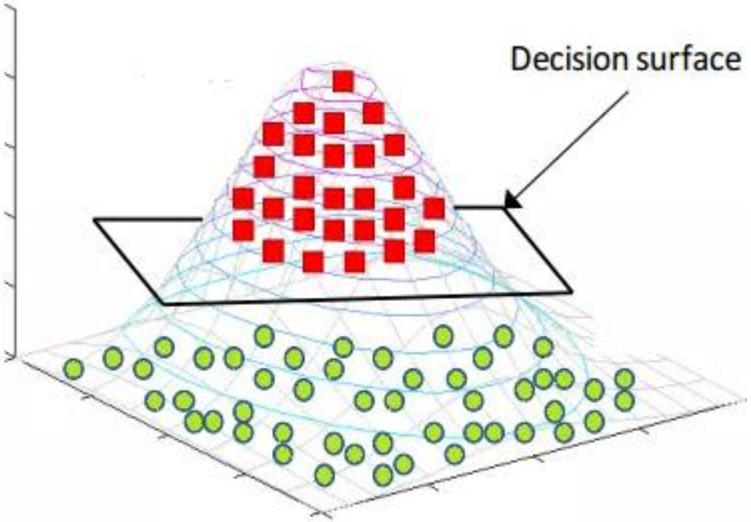
$$C = \frac{1}{\alpha}$$

Support Vector Machines

Some times data is not separable with a line / hyperplane



The **kernel** trick

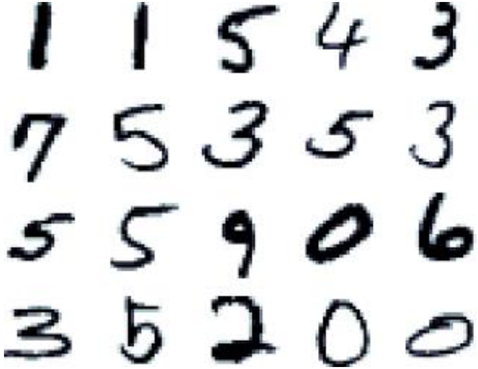


Visualization from http://blog.csdn.net/sinat_35257860/article/details/58226823

Support Vector Machine applications

Face Detection

Spam Filter



Handwriting recognition

Support Vector Machine applications

NASA EO-1

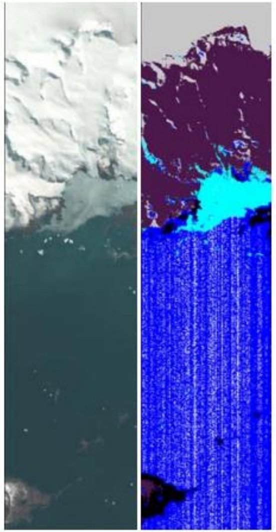


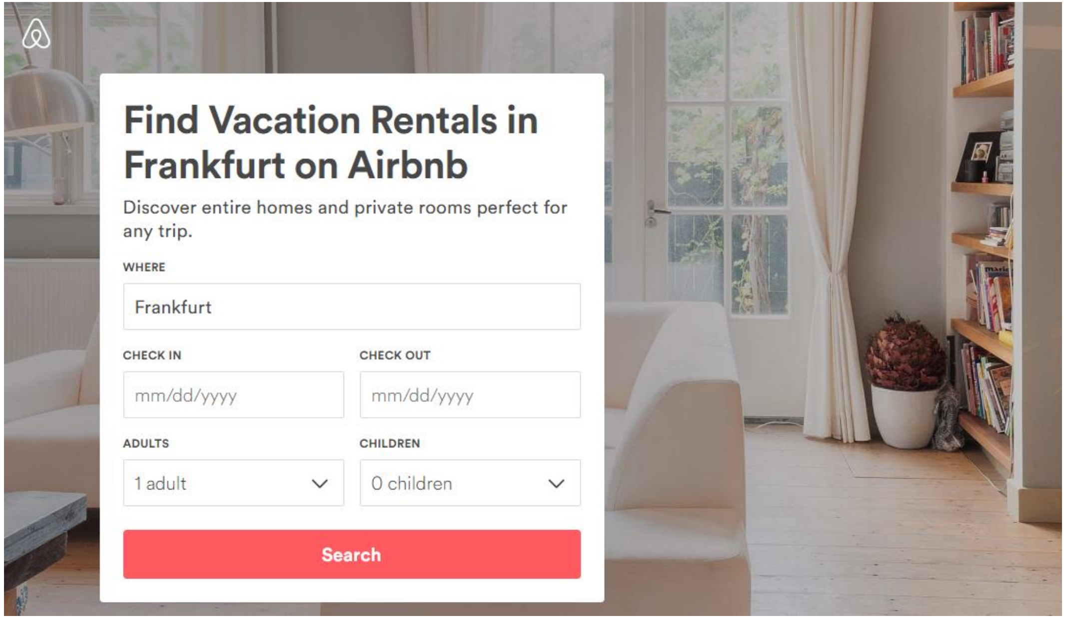
Figure 5. Image of South Georgia Island near Antarctica taken December 1, 2004. The left is the false color image while the right shows the resulting SVM classification, where blue is water, black is land, cyan is ice, purple is snow, gray is cloud, and white is unclassified. Open water was correctly identified indicating sea ice break-up and triggering another image of the scene to be taken on December 3, 2004.

Figure 4. Image of Lake Winnibigoshish, Wisconsin taken September 22, 2004. The scene was correctly classified as cloudy by the onboard SVM classifier.

Castano, Rebecca, Dominic Mazzoni, Nghia Tang, Ron Greeley, Thomas Doggett, Ben Cichy, Steve Chien, and Ashley Davies. "Onboard classifiers for science event detection on a remote sensing spacecraft." In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 845-851. ACM, 2006.

Python Hands-On

Imagine you had an apartment in Frankfurt and you want to use Airbnb to monetize it. What price should you ask?



Materials: Slides, Code, Videos

They will be available on the Data Analytics ESA connect community

url: https://connect.esa.int/communities/community/data-analytics

For externals, I'll post them on LinkedIn:

https://www.linkedin.com/in/josemartinezheras/

What is next?

March 14th 16:00 – HI

Session 3: Supervised Learning (2)

- Decision Trees
- Ensembles
- Random Forests

- Hands on

Resources

Watch the video of this lecture

https://dlmultimedia.esa.int/download/public/videos/2048/03/004/4803_004_AR_EN.mp4

Watch the practical exercise video

https://dlmultimedia.esa.int/download/public/videos/2048/03/003/4803_003_AR_EN.mp4

Get presentation and additional resources on

https://github.com/jmartinezheras/2018-MachineLearning-Lectures-ESA

Thank you

Data Analytics Team for Operations (DATO)

Jose Martinez Heras

LinkedIn: https://www.linkedin.com/in/josemartinezheras/