Lecture 5

Logistic Regression

GEOL 4397: Data analytics and machine learning for geoscientists

Jiajia Sun, Ph.D.

Feb. 5th, 2018

UNIVERSITY of HOUSTON

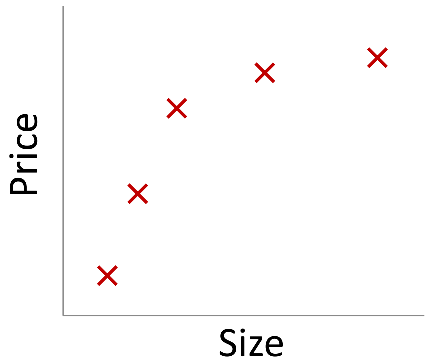
YOU ARE THE PRIDE

EARTH AND ATMOSPHERIC SCIENCES

Today's agenda

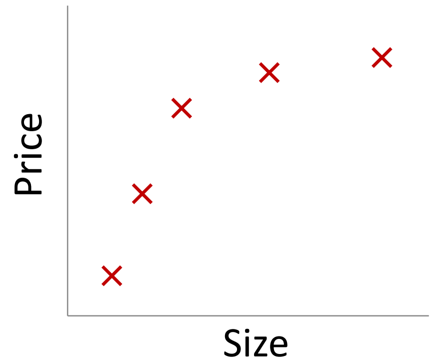
- Recap: diagnosing bias vs. variance
- Logistic regression: idea
- Understanding logistic regression
- Cost function
- Implementation using Scikit-Learn
  - Iris example data
  - Seismic receiver functions

Bias/variance



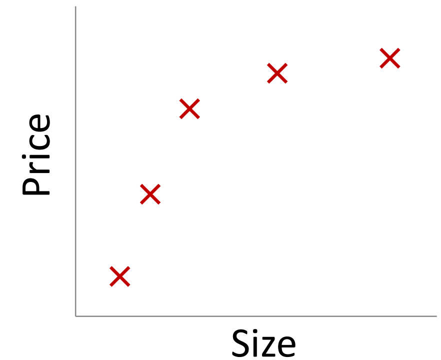
$\theta_0 + \theta_1 x$

High bias (underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance (overfit)

This slide is taken from Andrew Ng's ML class on coursera

Bias vs. Variances

Bias

- Due to over-simplified assumptions
- E.g., assuming a linear model when the training data are actually from a non-linear model
- Lead to underfitting the training data

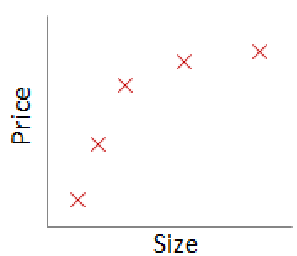
Variance

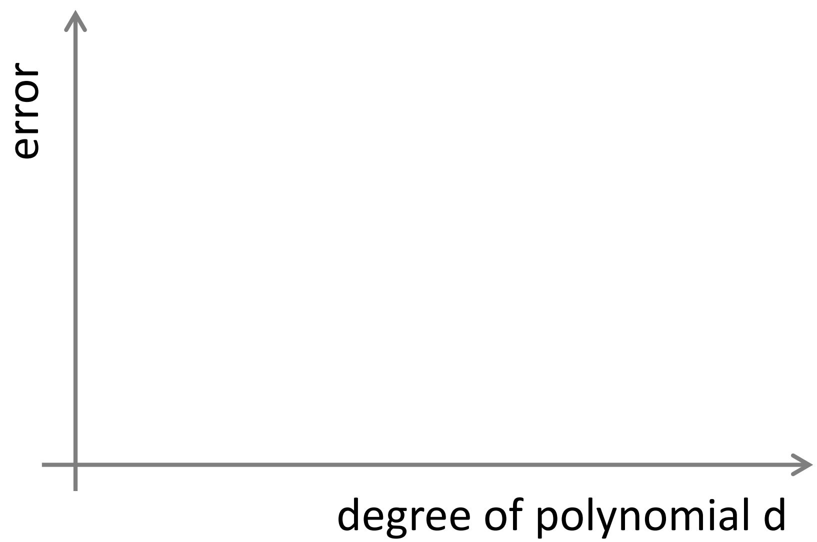
- Due to your model’s excessive sensitivity to small variations in the training data
- E.g., assume a highly nonlinear model when the data are actually linear
- Lead to overfitting the data

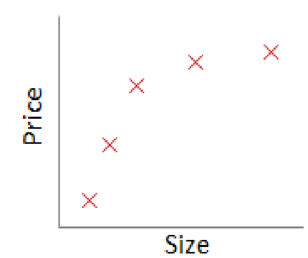
Bias/variance

Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

Validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



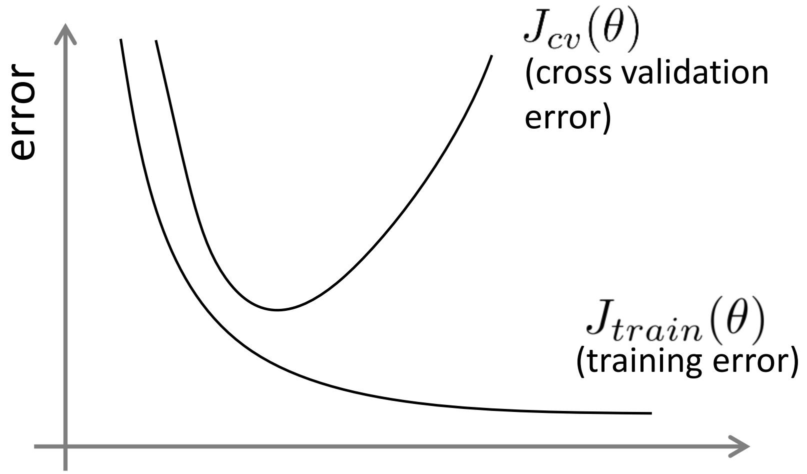




This slide is taken from Andrew Ng's ML class on coursera

Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than you were hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high.) Is it a bias problem or a variance problem?



degree of polynomial d
number of layers in NN
depth in a decision tree
degree of regularization

Bias (underfit):

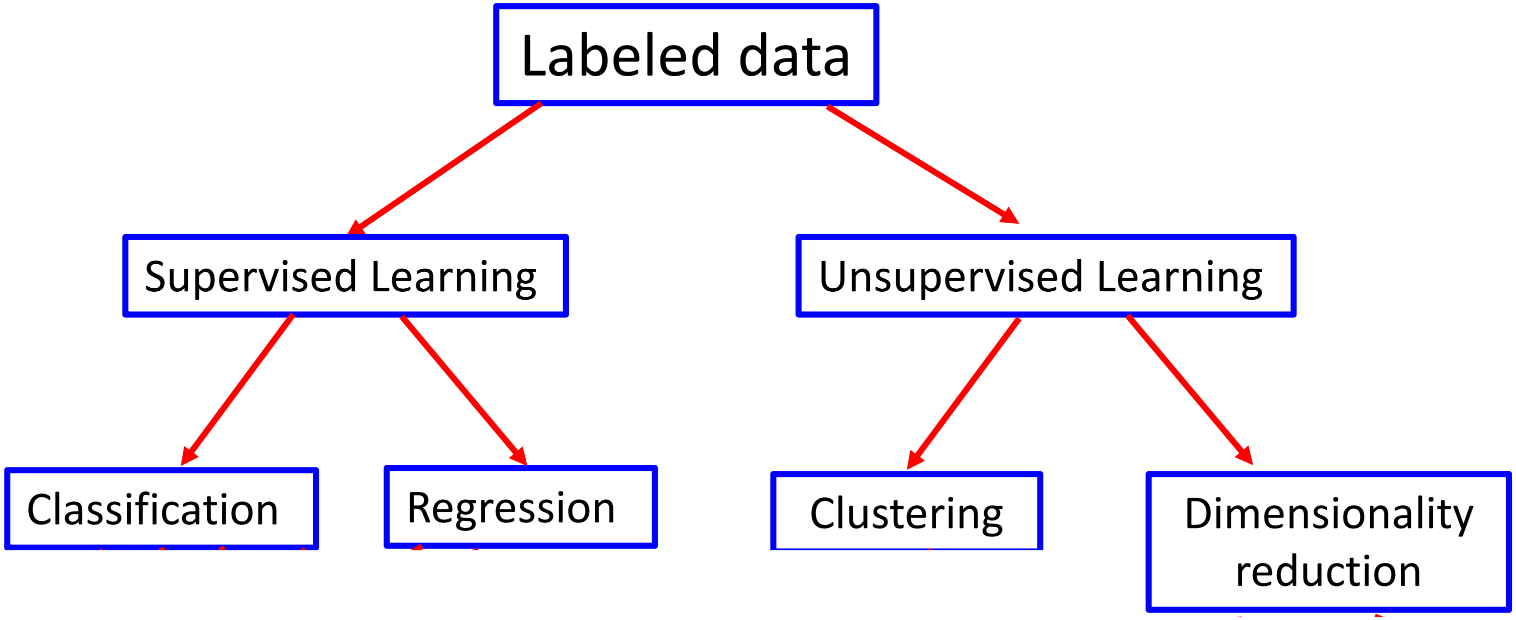
Variance (overfit):

This slide is taken from Andrew Ng's ML class on coursera

Today’s agenda

- Recap: diagnosing bias vs. variance
- Logistic regression: idea
- Understanding logistic regression
- Cost function
- Implementation using Scikit-Learn
  - Iris example data
  - Seismic receiver functions

Machine learning algorithms



Logistic regression: basic concept

- This is a classification method
- The name *logistic regression* is very confusing.

Logistic regression: applications

- Email: Spam (Yes / No)
- Online transactions: Fraudulent (Yes / No)
- Tumor: Malignant / Benign
- Object detection: cat (Yes / No), pedestrian (Yes / No)

- Salt body detection: Salt (Yes / No)
- Seismic trace QC: quality (good / bad)

Supervised learning

- Supervised learning is all about learning a mapping function from the input, x, to the output, y:

$$y = f(x)$$

- So that, given a new instance (e.g., a new email, a new image, a new transaction, etc), x, your model learning model can predict its category y.

Logistic regression: output variables

$$y = \{0, 1\}$$

0: ‘Negative class’ (e.g., not salt)

1: ‘Positive class’ (e.g., salt)

Linear regression

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- $h_\theta(x)$: life satisfaction
- x : GDP per capita
- θ_0 , θ_1: model parameters (to be learned from training data)

Linear regression for one feature

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- $h_\theta(x)$: life satisfaction
- x : GDP per capita
- θ_0 , θ_1: model parameters (to be learned from training data)

- Feature = input variable
- One feature = one input variable

Linear regression for multiple features

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $h_\theta(x)$: life satisfaction
- x_1 : GDP per capita x_2: medical care x_3 ...
- $\theta_0, \theta_1, \ldots \theta_n$: model parameters (to be learned from training data)

- Feature = input variable
- One feature = one input variable
- Multiple features = multiple input variables

Linear regression for multiple features

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $h_\theta(x)$: predictions
- x_1 : feature 1 x_2: feature 2 x_3 ...
- $\theta_0, \theta_1, \ldots \theta_n$: model parameters (to be learned from training data)

- Feature = input variable
- One feature = one input variable
- Multiple features = multiple input variables

Linear regression for multiple features

$$h_\theta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \dots \\ \theta_n \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \qquad x_0 = 1$$

- Feature = input variable
- One feature = one input variable
- Multiple features = multiple input variables

Output

- For logistic regression, we would like the output $h_\theta(\boldsymbol{x})$ to be {0, 1}, or

$$0 \le h_\theta(\boldsymbol{x}) \le 1$$

- The output of a linear regression model could be any real number > 1 or < 0

Logistic function

$$g(x) = \frac{1}{1 + e^{-x}}$$

Also called sigmoid function

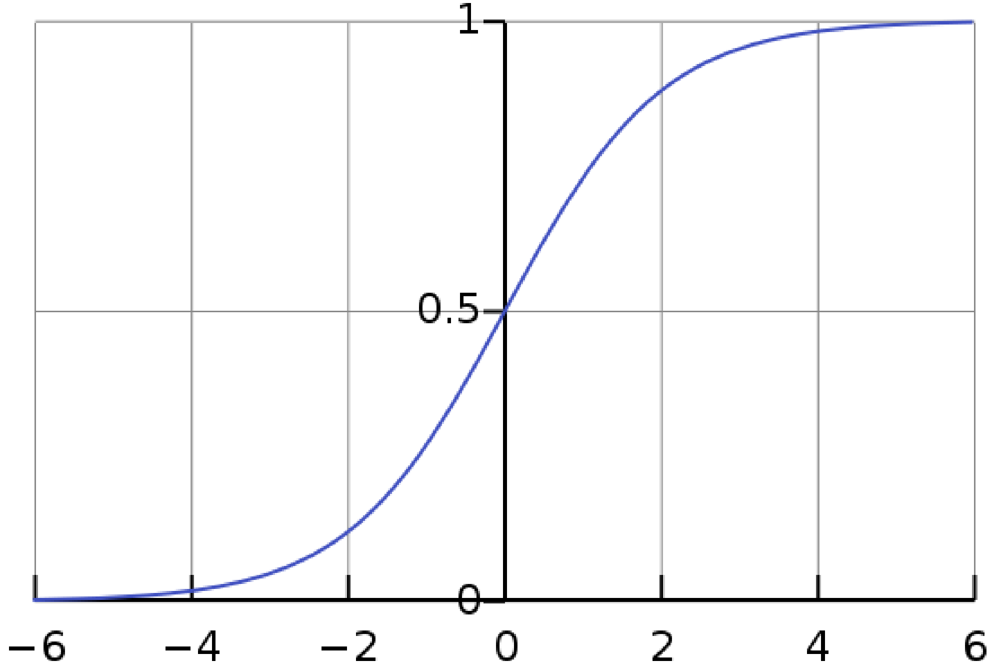
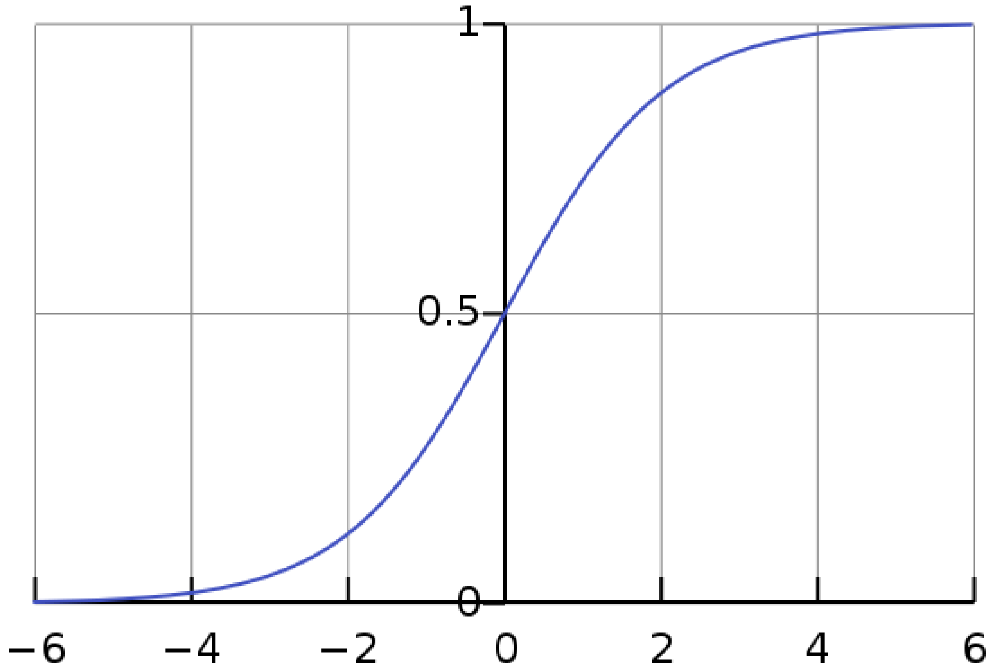


Image source: https://en.wikipedia.org/wiki/Logistic_function

Logistic function

$$g(x) = \frac{1}{1 + e^{-x}}$$

Also called sigmoid function



- Map any real number in $[-\infty, +\infty]$ to a real number within [0, 1]

Image source: https://en.wikipedia.org/wiki/Logistic_function

From linear regression to logistic regression

$$h_\theta(\boldsymbol{x}) = \quad \boldsymbol{\theta}^T \boldsymbol{x}$$

From linear regression to logistic regression

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x})$$

- The output is strictly within [0, 1]
- We can easily interpret the output variable as the probability

From linear regression to logistic regression

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x})$$

- The output is strictly within [0, 1]
- We can easily interpret the output variable as the probability

Interpretation

- $h_\theta(\boldsymbol{x})$ = estimated probability that y = 1 on input $\boldsymbol{x}$

Example: If $\boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \#kw \end{bmatrix}$

4U
Amazing
Free money
Get it now
Guarantte
Cash
…

$h_\theta(\boldsymbol{x})$ = 0.82

82% chance that this email is spam

Classification

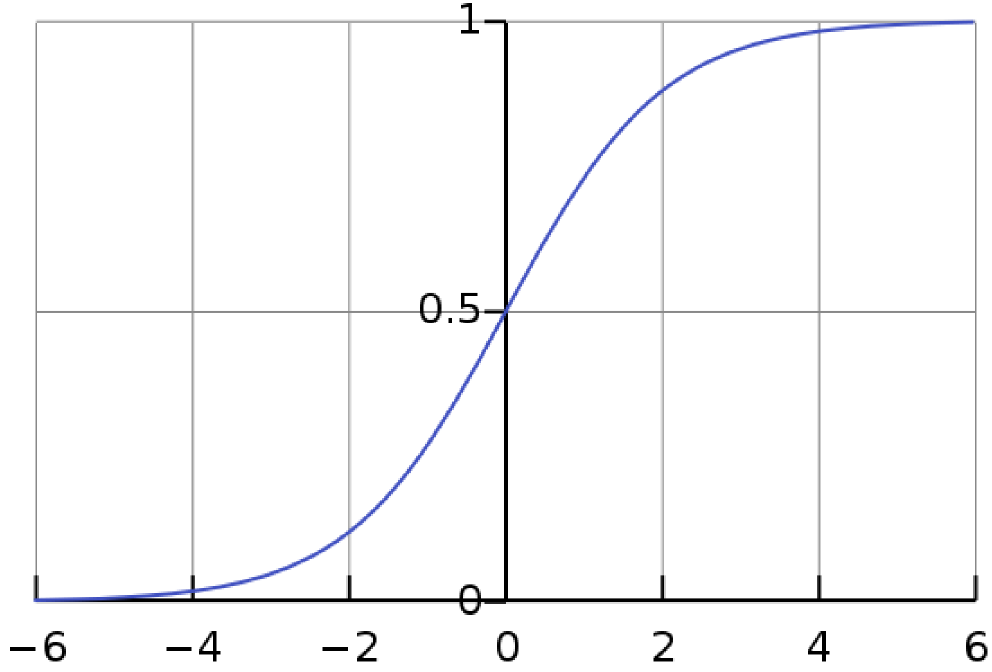
- If $h_\theta(\boldsymbol{x}) \geq 0.5$, predict $y = 1$
- If $h_\theta(\boldsymbol{x}) < 0.5$, predict $y = 0$

Today’s agenda

- Logistic regression: idea
- Understanding logistic regression
- Cost function
- Implementation using Scikit-Learn

Classification

- If $h_\theta(\boldsymbol{x}) \geq 0.5$, predict $y = 1$
- If $h_\theta(\boldsymbol{x}) < 0.5$, predict $y = 0$

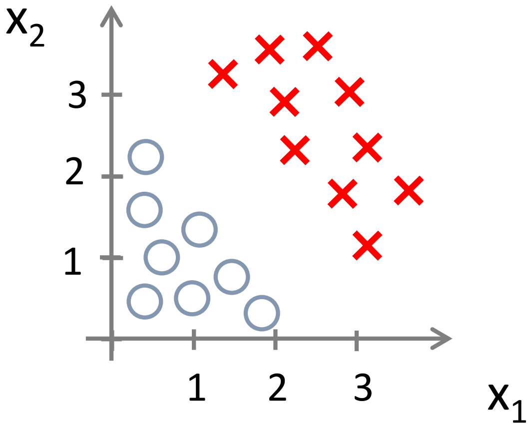


Classification

- If $\boldsymbol{\theta}^T \boldsymbol{x} \geq 0$, predict $y = 1$
- If $\boldsymbol{\theta}^T \boldsymbol{x} < 0$, predict $y = 0$

Understanding logistic regression

Decision Boundary

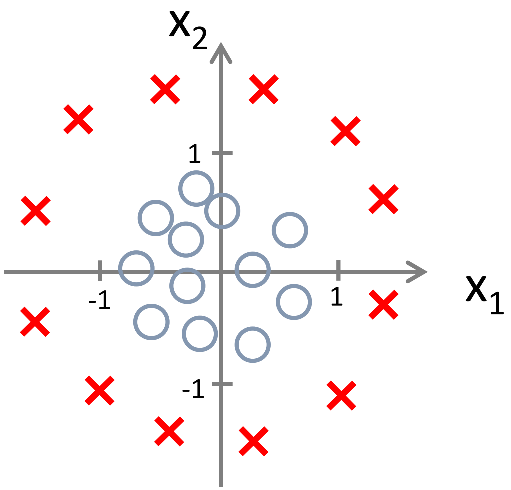


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

This slide is taken from Andrew Ng's ML class on coursera

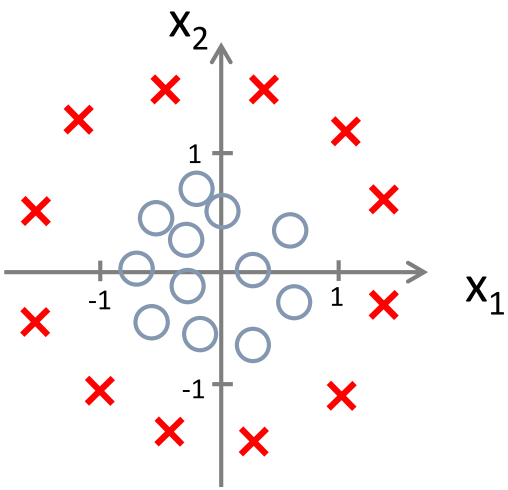
Nonlinear decision boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$$

This slide is taken from Andrew Ng's ML class on coursera

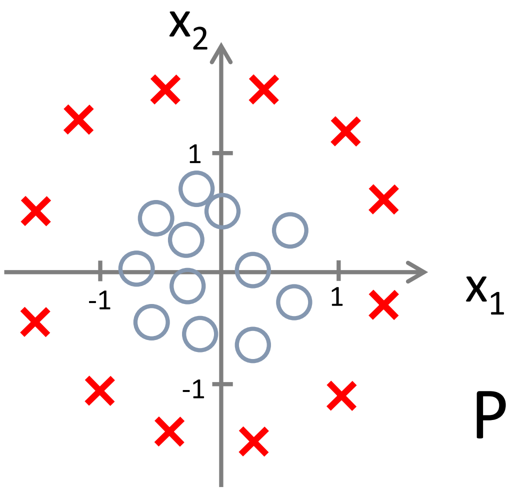
Nonlinear decision boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

This slide is taken from Andrew Ng's ML class on coursera

Nonlinear decision boundaries

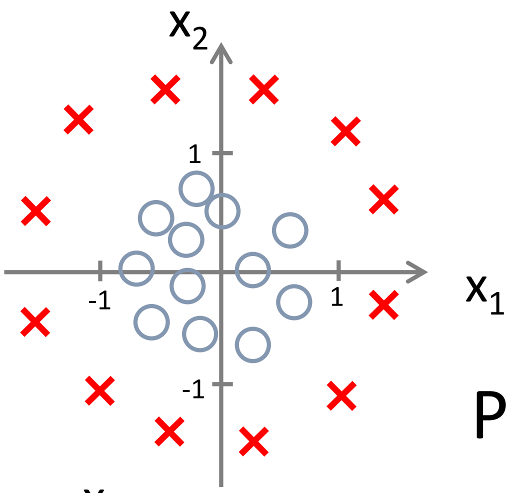


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

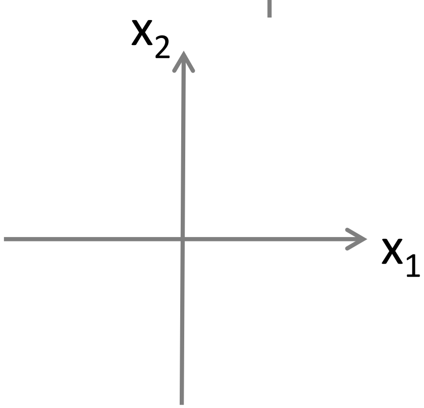
This slide is taken from Andrew Ng's ML class on coursera

Nonlinear decision boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \ldots)$$

This slide is taken from Andrew Ng's ML class on coursera

Remedy for underfitting

- Underfitting happens when your ML model is overly simple
- Therefore, possible solutions are:
  1. ~~Collect more training data~~
  2. ~~Reduce data noise~~
  3. Make your model more complex
     - using a high-degree polynomial model rather than a linear model
     - using less regularization
     - Adding more features such as $(x_1^2, x_2^2, x_1 x_2)$ to the learning algorithm (feature engineering)

Today’s agenda

- Recap: diagnosing bias vs. variance
- Logistic regression: idea
- Understanding logistic regression
- Cost function
- Implementation using Scikit-Learn
  - Iris example data
  - Seismic receiver functions

Logistic regression: cost function

- Training set

$$\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$$

- m examples
- Each example has n features.

$$\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$$

Note

- Materials from Slide 38 to 49 explain how to develop the cost function for logistic regression.
- They are beyond the scope this class. Feel free to skip them.
- These materials might be useful for those who want to learn more about cost function.

Logistic regression: cost function

Optional materials

- Training set

$$\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$$

Logistic regression: cost function

Optional materials

- Training set

$$\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$$

$$h_\theta(\boldsymbol{x}^{(1)}) \quad h_\theta(\boldsymbol{x}^{(2)}) \quad h_\theta(\boldsymbol{x}^{(m)})$$

Logistic regression: cost function

Optional materials

- Training set

$$\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$$

$$h_\theta(\boldsymbol{x}^{(1)}) \qquad h_\theta(\boldsymbol{x}^{(2)}) \qquad h_\theta(\boldsymbol{x}^{(m)})$$

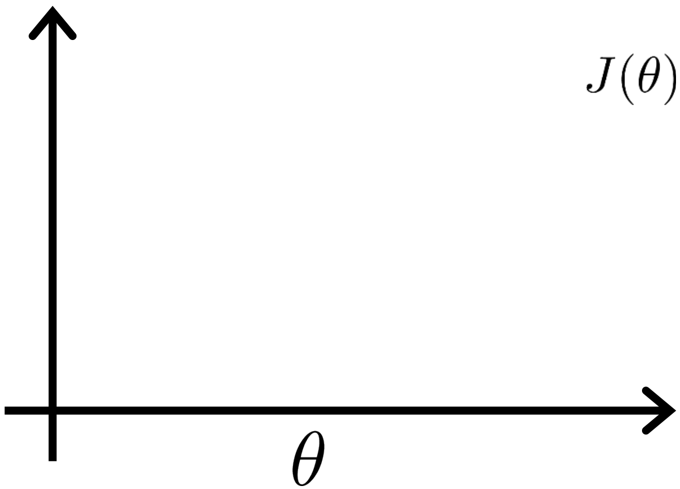
- Cost function

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta\left(\boldsymbol{x}^{(i)}\right) - y^{(i)})^2$$

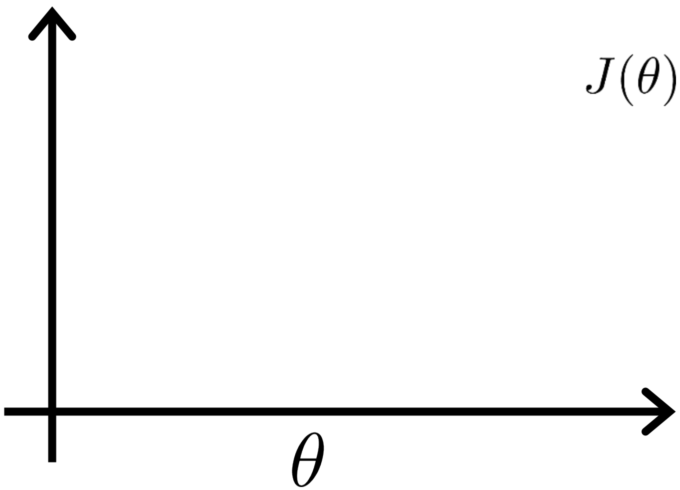
Non-convex vs convex function

Optional materials

Non-convex



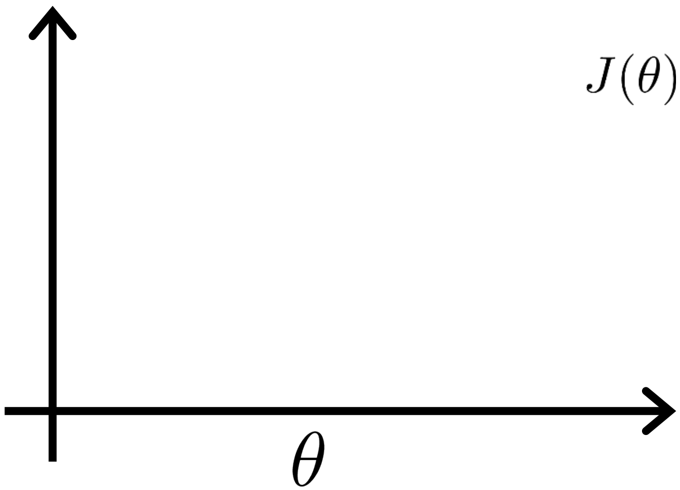
Convex



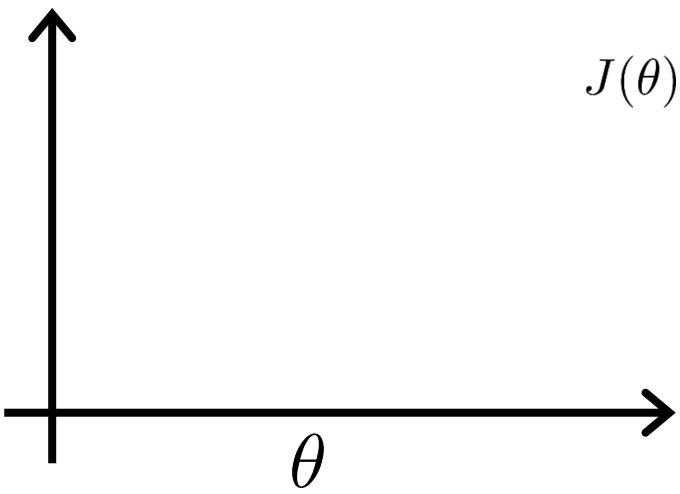
Non-convex vs convex function

Optional materials

Non-convex



Convex



For optimization, if ever possible, we would like to work with convex cost function.

Developing convex cost function for logistic regression

Optional materials

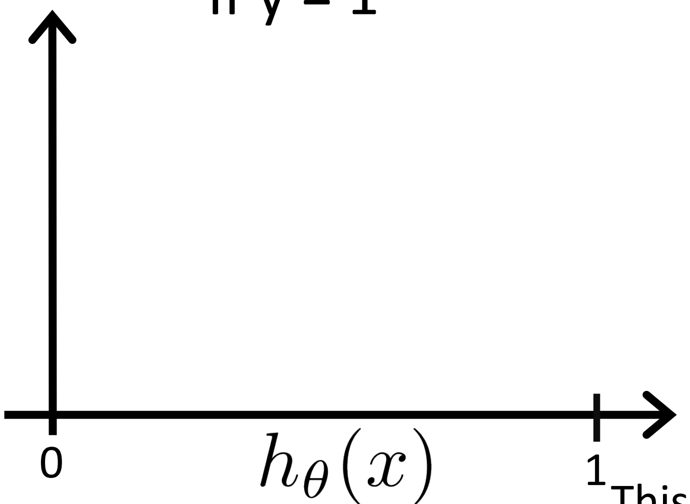
- Let us consider one single training example $\boldsymbol{x}$, and the associated label y
- Basic idea is that, if prediction $h_\theta(\boldsymbol{x})$ is very different from the true label y, we penalize the prediction heavily.
- Conversely, if the prediction $h_\theta(\boldsymbol{x})$ is close to the true label y, we penalize the prediction less.

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 1



$\mathrm{Cost} = 0 \text{ if } y = 1, h_\theta(x) = 1$

But as $h_\theta(x) \rightarrow 0$

$Cost \rightarrow \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

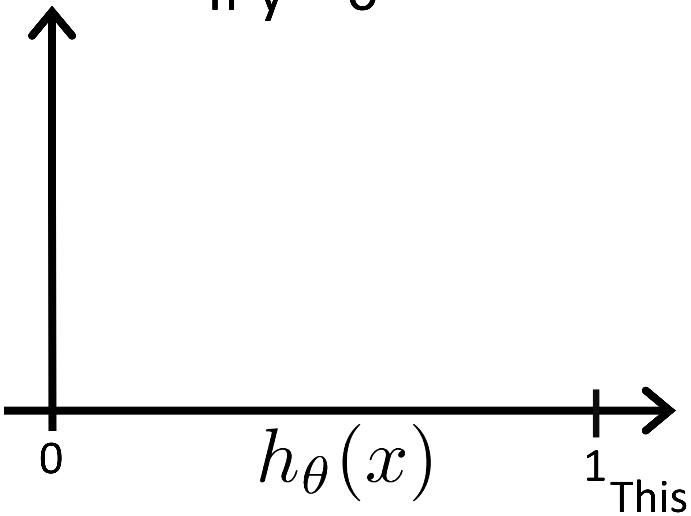
This slide is taken from Andrew Ng's ML class on coursera

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 0



This slide is taken from Andrew Ng's ML class on coursera

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$= -y \log\big(h_\theta(\boldsymbol{x})\big) - (1 - y) \log\big(1 - h_\theta(\boldsymbol{x})\big)$$

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$= -y\log\big(h_\theta(\boldsymbol{x})\big) - (1-y)\log\big(1 - h_\theta(\boldsymbol{x})\big)$$

Note: We have only considered one training instance.

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y\log\big(h_\theta(\boldsymbol{x})\big) - (1-y)\log\big(1 - h_\theta(\boldsymbol{x})\big)\right]$$

Logistic regression cost function

Optional materials

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$= -y\log\big(h_\theta(\boldsymbol{x})\big) - (1-y)\log\big(1 - h_\theta(\boldsymbol{x})\big)$$

Note: We have only considered one training instance.

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y\log\big(h_\theta(\boldsymbol{x})\big) - (1-y)\log\big(1 - h_\theta(\boldsymbol{x})\big)\right]$$

This is convex!!

Learning

- Minimize

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_\theta\left(\boldsymbol{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(\boldsymbol{x}^{(i)}\right)\right)\right]$$

- And obtain optimal $\boldsymbol{\theta}$

Learning/training

- Note that the cost function $J(\boldsymbol{\theta})$ is differentiable.
- It is straightforward to calculate the gradient

$$\nabla J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \\ \cdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix}$$

- Batch gradient descent, SGD, MGD are handy tools to do the training.

Making predictions

- Once when learning is completed, we would obtain the optimal model parameters $\boldsymbol{\theta}$
- Given a new data, $\boldsymbol{x}$, we can predict whether it belongs to positive or negative class by computing

$$h_\theta(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

Today's agenda

- Logistic regression: idea
- Understanding logistic regression
- Cost function
- Implementation using Scikit-Learn

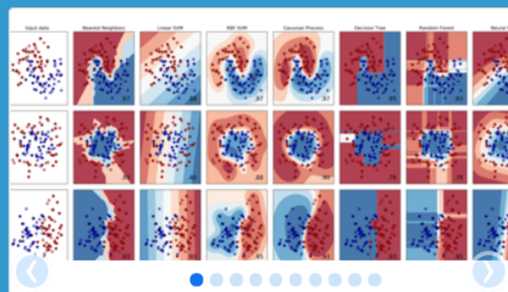
Scikit-Learn

https://scikit-learn.org/stable/index.html

scikit learn

Home | Installation | Documentation | Examples

Google Custom Search



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

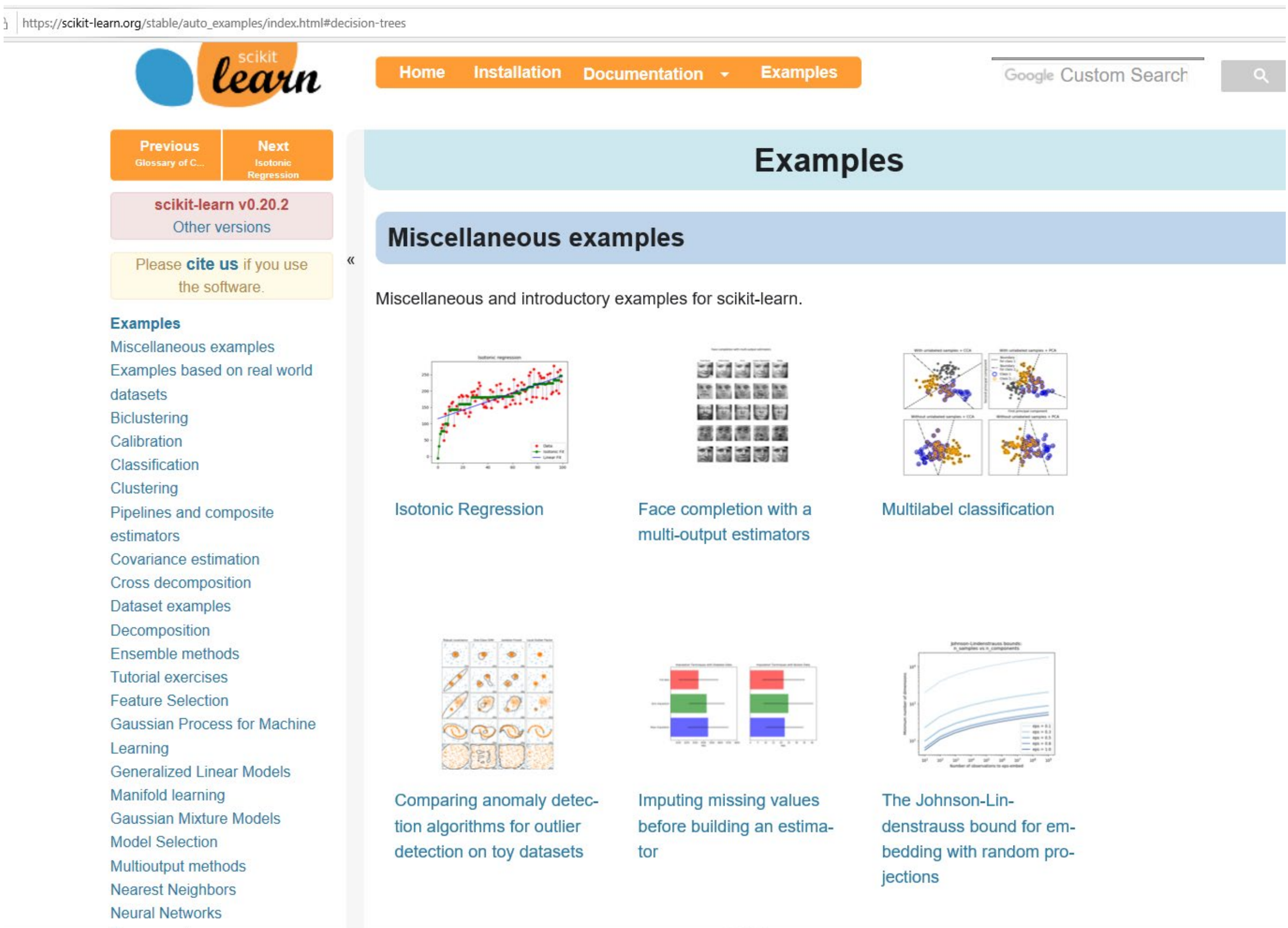
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

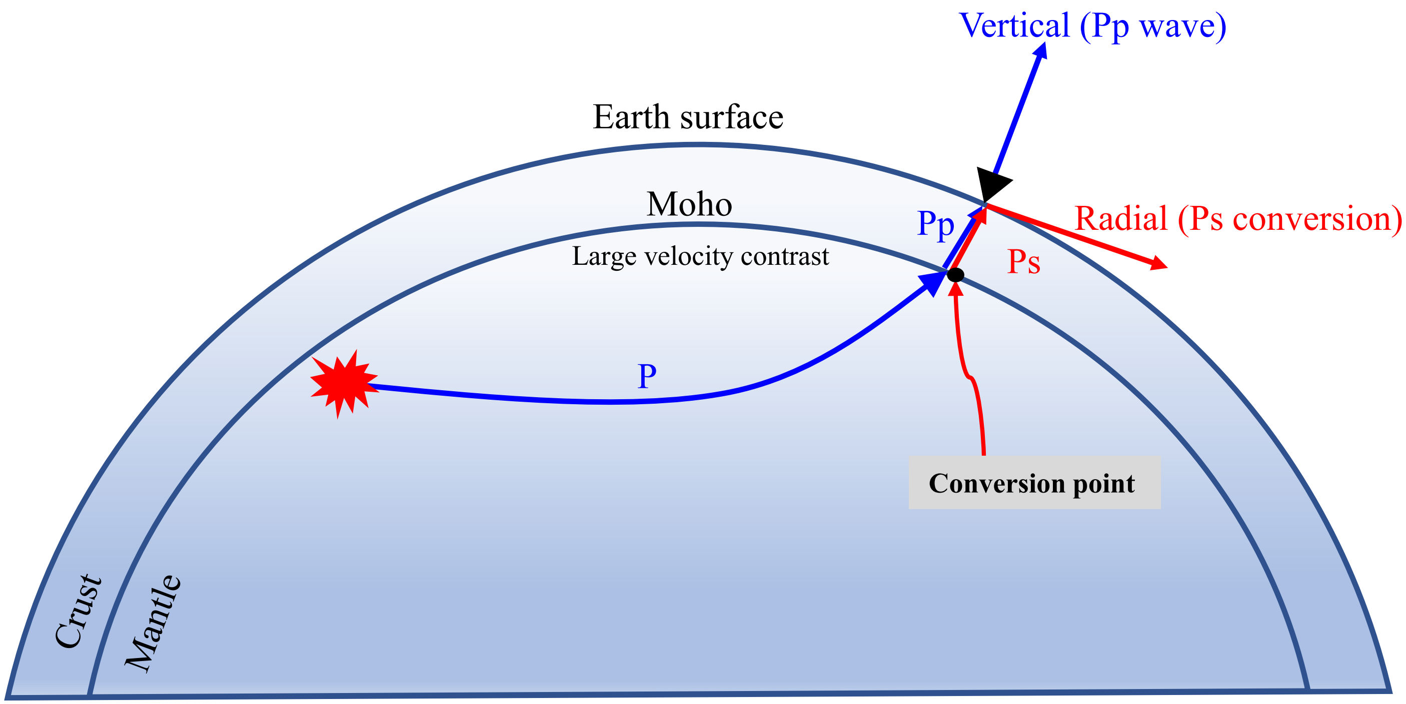
Scikit-Learn examples



Demonstration in Jupyter Notebook

Lab exercise: classifying receiver functions into two categories

Method



Receiver functions (RF) reflects the responses to the structure beneath the receiver by **calculating the conversion coefficient.**

Image credit: Ying Zhang from UH/EAS

Receiver functions

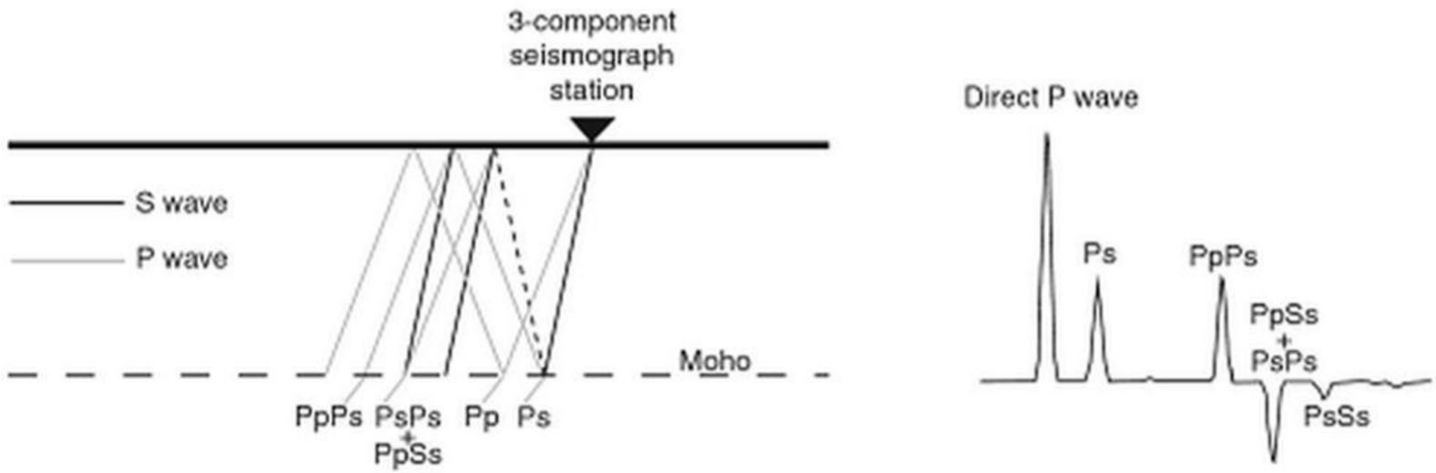
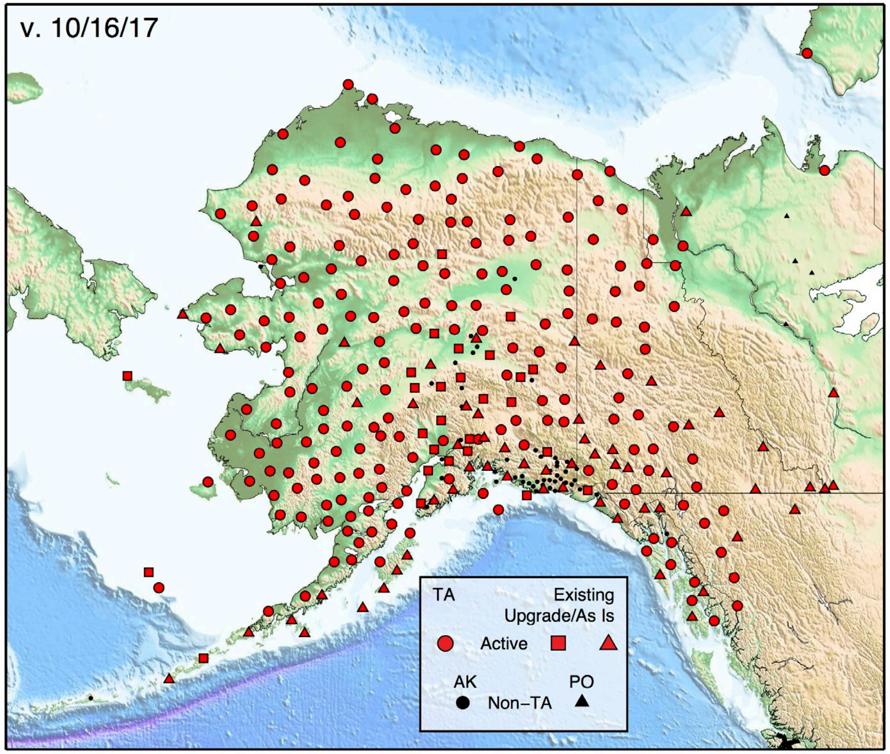


Figure 4 - Receiver function analysis of Earth structure beneath a seismograp station (left). The arrivals, and subsequent reverberations, can be isolated by receiver function analysis (right). Modified by Ammon (1991). (Courtesy of Hellfrich et. al, 2013)

https://www.duo.uio.no/bitstream/handle/10852/45482/Master_Thesis_Torsvik_Receiver_Function_Analysis.pdf?sequence=15

Our data

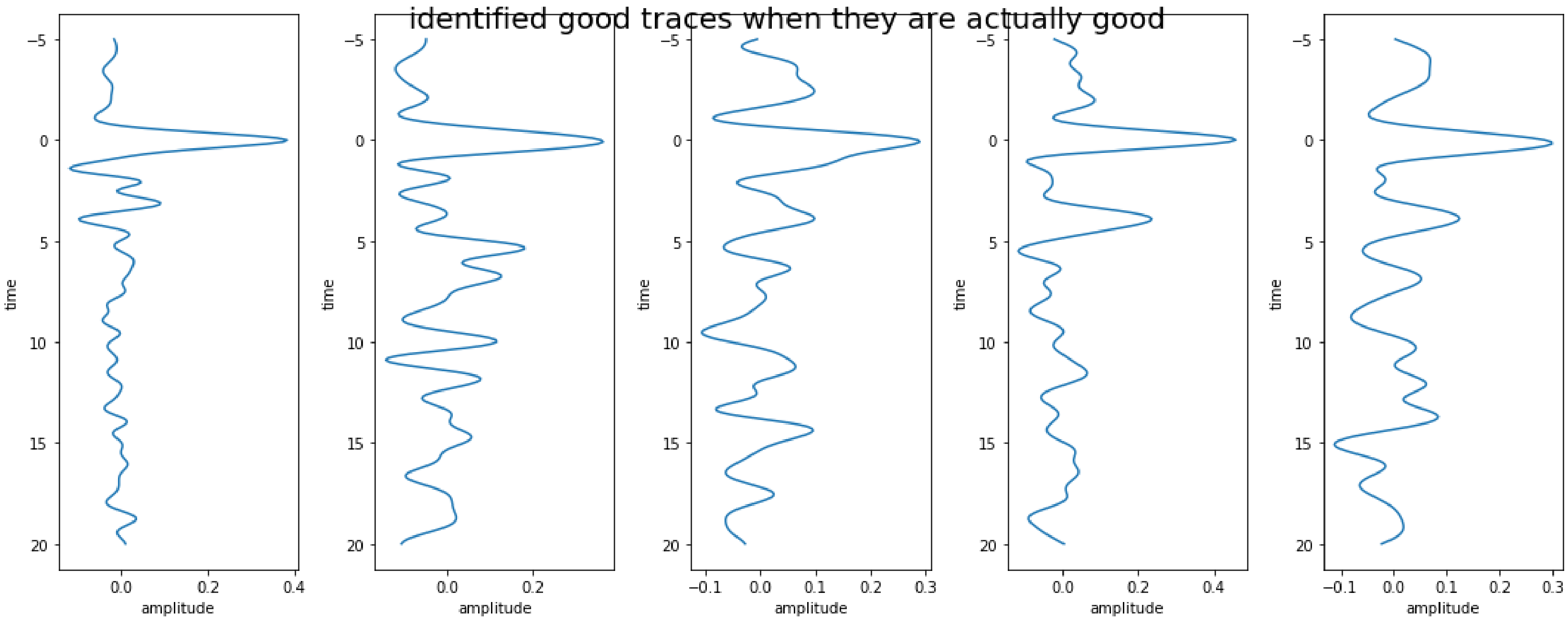
- P-wave receiver functions computed USArray TA and Alaska Regional Network



Our data

- P-wave receiver functions computed USArray TA and Alaska Regional Network
- We used the data from teleseismic P waves with body wave magnitudes (Mb) larger than 5.7 and epicentral distances within 30°–90° from the IRIS [*Zhang, 2017*].
- We obtained **12,597** P-wave receiver functions by filtering, rotating and deconvolving the recorded raw seismic data.
- Manually classified as ‘good’ and ‘bad’.

Our receiver functions



Our receiver functions