Logistic Regression

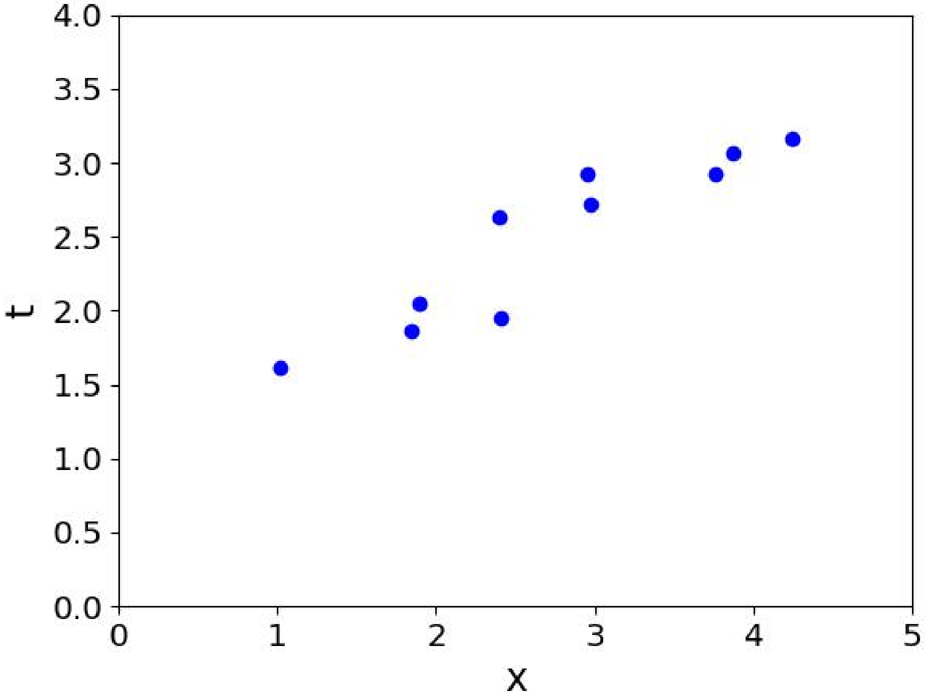
Zhu Suguo

2021.03.23

Linear Regression - Setup

In supervised learning:

- There is input $\mathbf{x} \in \mathcal{X}$, typically a vector of features (or covariates)
- There is target $t \in \mathcal{T}$ (also called response, outcome, output, class)
- Objective is to learn a function $f : \mathcal{X} \to \mathcal{T}$ such that $t \approx y = f(\mathbf{x})$ based on some data $\mathcal{D} = \{(\mathbf{x}^{(i)}, t^{(i)}) \text{ for } i = 1, 2, ..., N\}$.



Linear Regression - Model

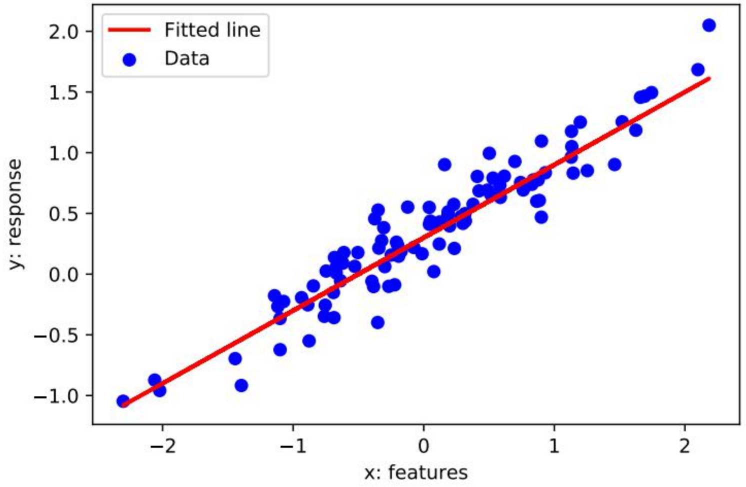
- Model: In linear regression, we use a *linear* function of the features $\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{R}^D$ to make predictions y of the target value $t \in \mathbb{R}$:

$$y = f(\mathbf{x}) = \sum_j w_j x_j + b$$

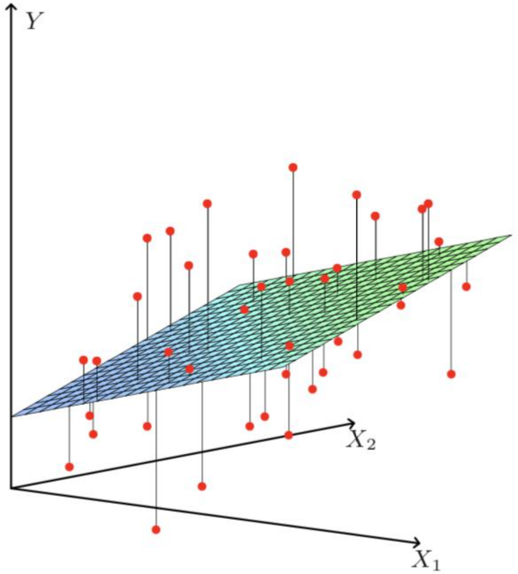
  - y is the prediction
  - $\mathbf{w}$ is the weights
  - b is the bias (or intercept)
- $\mathbf{w}$ and b together are the parameters
- We hope that our prediction is close to the target: $y \approx t$.

What is Linear? 1 feature vs D features

- If we have only 1 feature: $y = wx + b$ where $w, x, b \in \mathbb{R}$.
- y is linear in x.



- If we have D features: $y = \mathbf{w}^\top \mathbf{x} + b$ where $\mathbf{w}, \mathbf{x} \in \mathbb{R}^D$, $b \in \mathbb{R}$
- y is linear in $\mathbf{x}$.



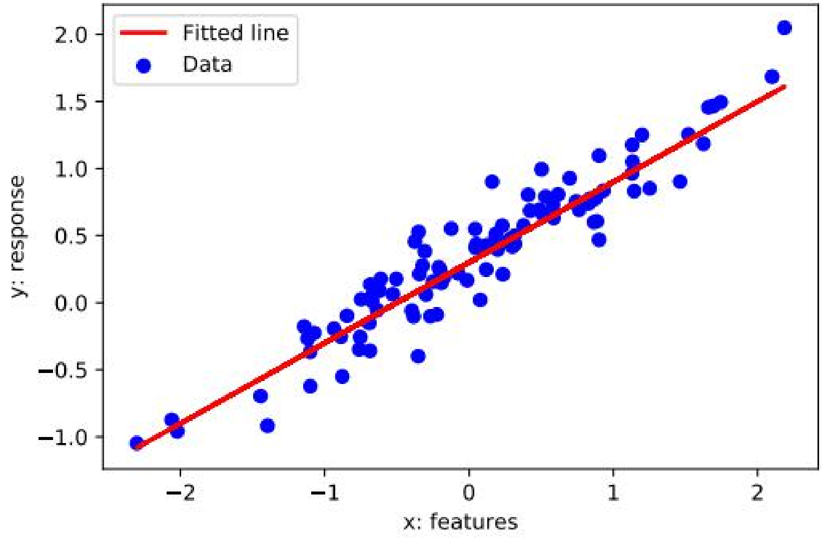
Relation between the prediction y and inputs x is linear in both cases.

Linear Regression

We have a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, t^{(i)})$ for $i = 1, 2, ..., N\}$ where,

- $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_D^{(i)})^\top \in \mathbb{R}^D$ are the inputs (e.g. age, height)
- $t^{(i)} \in \mathbb{R}$ is the target or response (e.g. income)
- predict $t^{(i)}$ with a linear function of $\mathbf{x}^{(i)}$:

  - $t^{(i)} \approx y^{(i)} = \mathbf{w}^\top \mathbf{x}^{(i)} + b$
  - Different $(\mathbf{w}, b)$ define different lines.
  - We want the "best" line $(\mathbf{w}, b)$.
  - How to quantify "best"?



Linear Regression - Loss Function

- A loss function $\mathcal{L}(y, t)$ defines how bad it is if, for some example $\mathbf{x}$, the algorithm predicts y, but the target is actually t.
- Squared error loss function:

$$\mathcal{L}(y, t) = \tfrac{1}{2}(y - t)^2$$

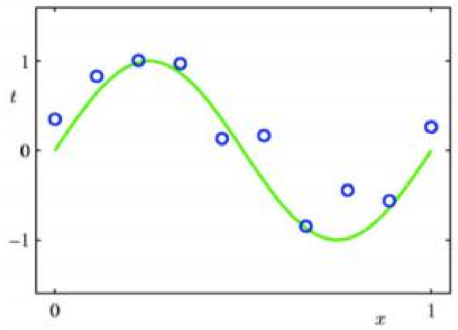
- $y - t$ is the residual, and we want to make this small in magnitude
- The $\frac{1}{2}$ factor is just to make the calculations convenient.
- Cost function: loss function averaged over all training examples

$$\begin{aligned}\mathcal{J}(\mathbf{w}, b) &= \frac{1}{2N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)^2 \\ &= \frac{1}{2N}\sum_{i=1}^{N}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - t^{(i)}\right)^2\end{aligned}$$

- Terminology varies. Some call “cost” *empirical* or *average loss*.

Polynomial Feature Mapping

If the relationship doesn't look linear, we can fit a polynomial.



Fit the data using a degree-M polynomial function of the form:

$$y = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{i=0}^{M} w_i x^i$$

- Here the feature mapping is $\boldsymbol{\psi}(x) = [1, x, x^2, \ldots, x^M]^\top$.
- We can still use linear regression to find $\mathbf{w}$ since $y = \boldsymbol{\psi}(x)^\top \mathbf{w}$ is linear in $w_0, w_1, \ldots$.
- In general, $\boldsymbol{\psi}$ can be any function. Another example: $\boldsymbol{\psi}(x) = [1, \sin(2\pi x), \cos(2\pi x), \sin(4\pi x), \ldots]^\top$.

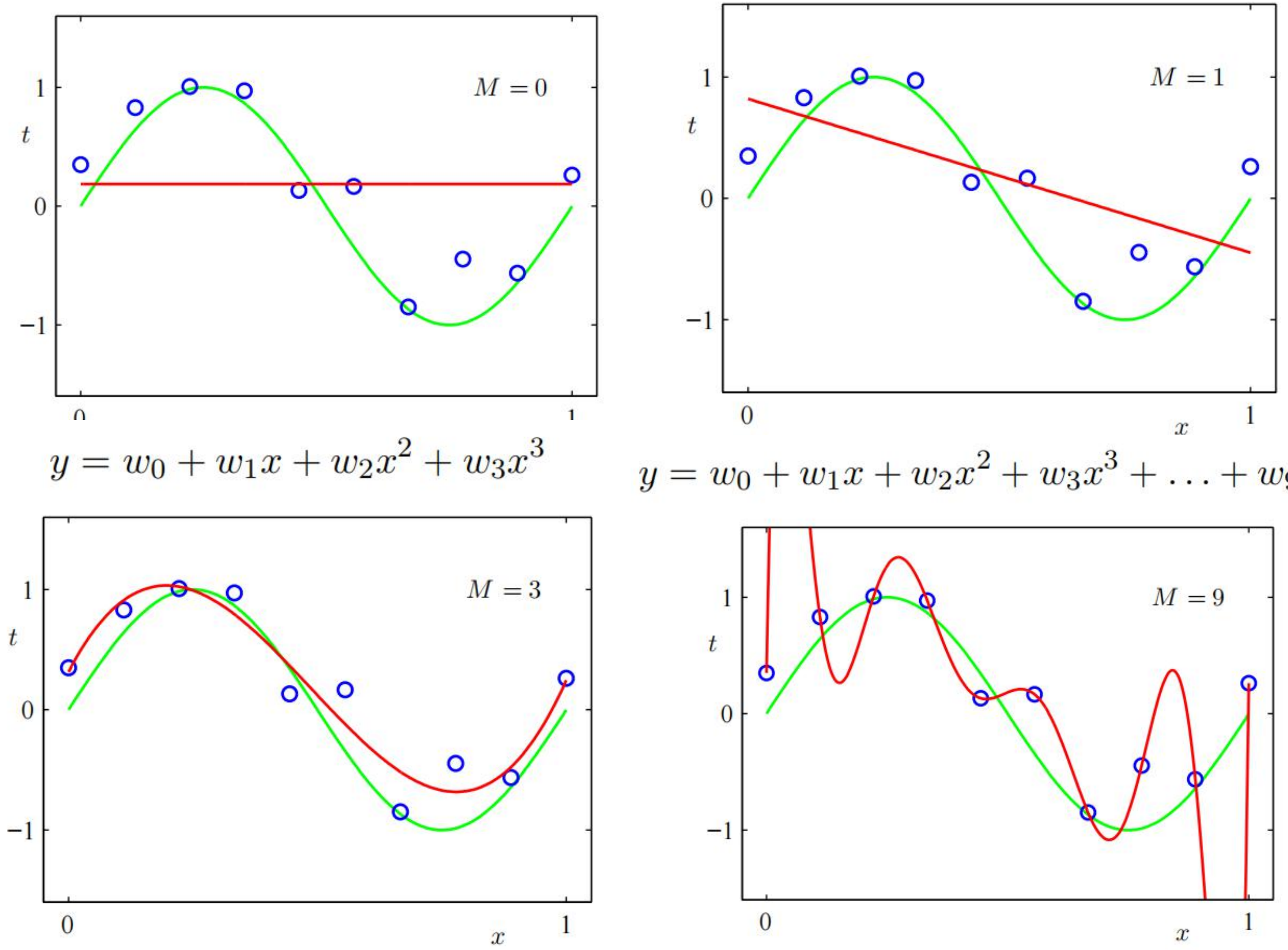
Polynomial Feature Mapping with M =0， 1， 3， 9

$$y = w_0$$

$$y = w_0 + w_1 x$$

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

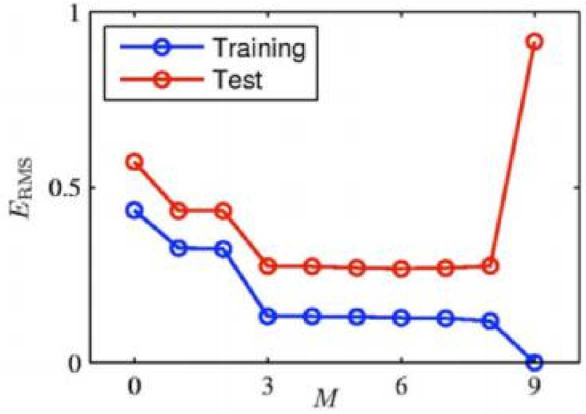
$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_9 x^9$$

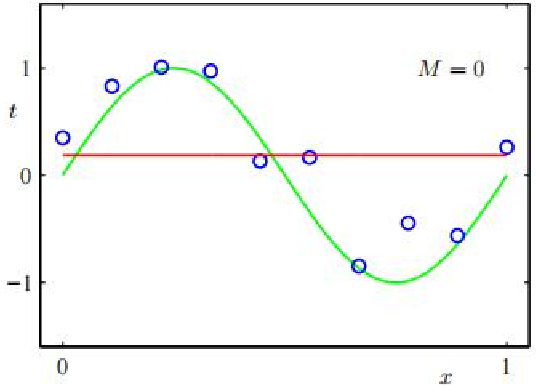


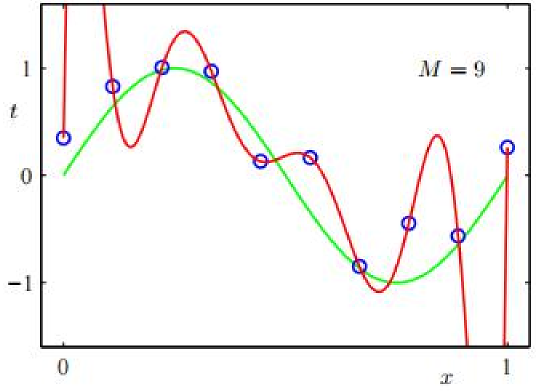
Model Complexity and Generalization

Underfitting (M=0): model is too simple — does not fit the data.

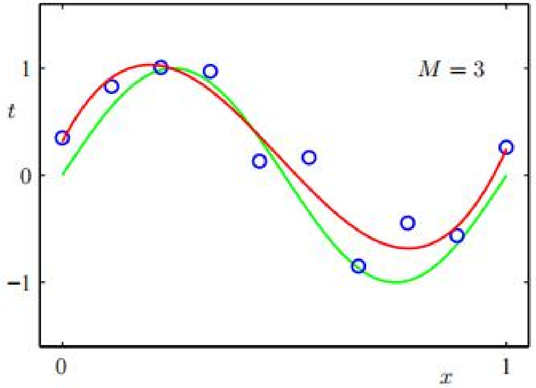
Overfitting (M=9): model is too complex — fits perfectly.







Good model (M=3): Achieves small test error (generalizes well).



Q & A

- What do you do with it when you have encountered a overfitting?
- or Which methods can deal with overfitting?

Syllabus

Syllabus

Classification

Supervised Learning

Label discrete

- ✓ Classification
  - ✓ **Tumour or not**: the size of the tumor and the age of the patient
  - ✓ **The credit card default or not**: the user's age, occupation, and number of deposits

The input variables can be either discrete or continuous。

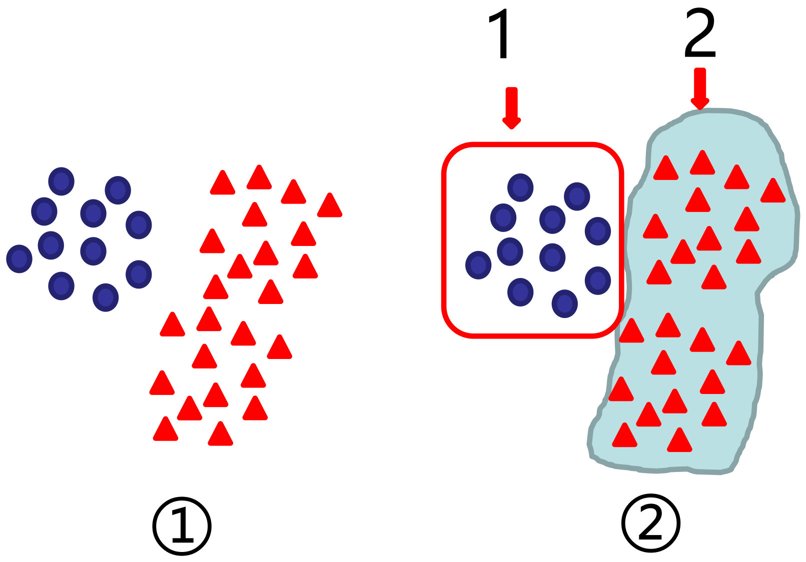
Classification

Binery classification

Sample 1: blue circle;

Sample 2: the others

①->②



Binery classification

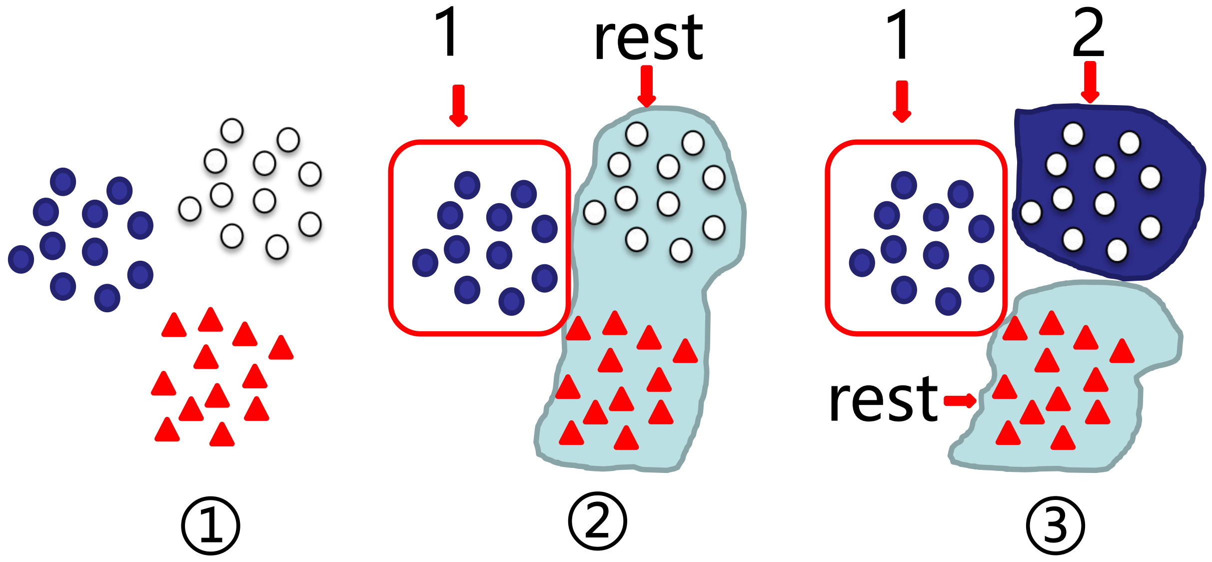
Classification

Milti-class classification

Positive class: one of the classes

Negative class: rest

①->②->③->……



One-vs-All (One-vs-Rest)

Syllabus

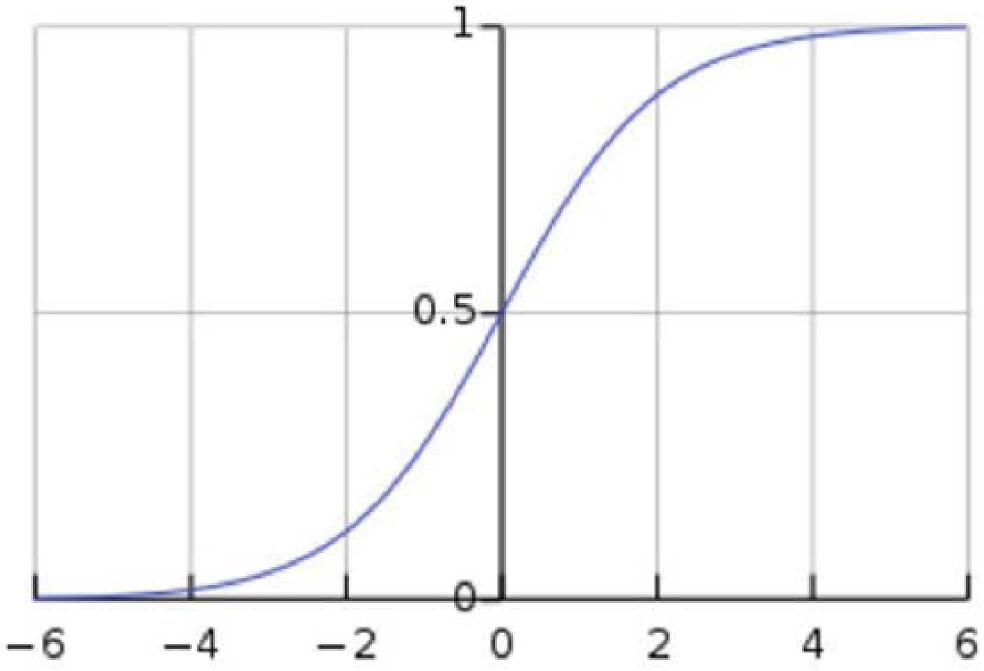
2.Sigmoid function

Sigmoid function

$\sigma(z)$: logistic function,

$g(z)$: Sigmoid function

$$\sigma(z) = g(z) = \frac{1}{1+e^{-z}}, \quad z=w^T x + b$$



Then, the hypothesis function of logistic regression:

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

when $\sigma(z)$>=0.5, predict y=1

when $\sigma(z)$<0.5, predict y=0

2.Sigmoid function

- $h(x) = z = w^T x$, $x \in (-\infty, +\infty)$, prediction results: $[0,1]$

In binery classification, the odds of experiencing an event: $\frac{p}{1-p}$, which is the ratio of the probability of no occurrence.

where p is the probability of a random event, and the range of p is $[0,1]$。

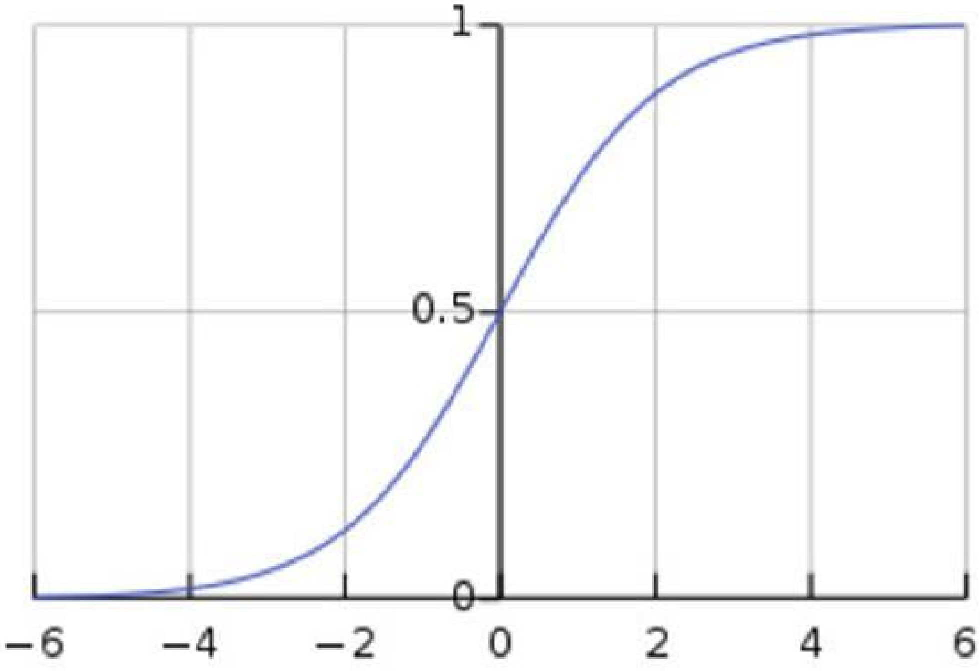
- $\log\frac{p}{1-p}$, while $\log\frac{p}{1-p} = w^T x = z$

$$p = \frac{1}{1+e^{-w^T x}} = \frac{1}{1+e^{-z}}$$

2.Sigmoid function

z logistic transformation： $g(z) = \frac{1}{1+e^{-z}}$

$$
\begin{aligned}
g'(z) &= \left(\frac{1}{1+e^{-z}}\right)' \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\
&= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z))
\end{aligned}
$$



$g(z)$

Syllabus

3.Logistic regression

Assum a binary classification model:

$$p(y=1|x;w)=h(x)$$

$$p(y=0|x;w)=1-h(x)$$

then:

$$p(y|x;w)=(h(x))^{y}(1-h(x))^{1-y}$$

Hypothesis of logistic regression: $h(x)=g(w^{T}x)=g(z)$

where $z=w^{T}x$, **logistic function** is:

$$g(z)=\frac{1}{1+e^{-z}},\quad g'(z)=g(z)(1-g(z))$$

3.Logistic regression

Loss function:

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$\hat{y}$ is $h(x)$

y it the true-value

Cost function:

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\hat{y}^{(i)} - (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$$

3.Logistic regression

Likelihood function: $L(w) = \prod_{i=1}^{m} P\,(y^{(i)}|x^{(i)}; w) = \prod_{i=1}^{m} (\,h(x^{(i)}))^{y^{(i)}}(1 - h(x^{(i)}))^{1-y^{(}}$

Log fucntion:

$$l(w) = \mathrm{log}L(w) = \sum_{i=1}^{m} (\,y^{(i)}\mathrm{log}(h(x^{(i)})) + (1 - y^{(i)})\mathrm{log}(1 - h(x^{(i)})))$$

Then, cost fucntion is :

$$J(w) =- \frac{1}{m} l(w) =- \frac{1}{m} \sum_{i=1}^{m} (\,y^{(i)}\mathrm{log}(h(x^{(i)})) + (1 - y^{(i)})\mathrm{log}(1 - h(x^{(i)})))$$

3.Logistic regression

Gradient descent：

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w}$$

$$J(w) = -\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right)$$

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

then： $w_j := w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$

3.Logistic regression

The derivation process:

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$J(w) = -\frac{1}{m} \sum_{i=1}^{m} (\underline{y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))})$$

$$\Downarrow$$

$$\begin{aligned}
& y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \\
&= y^{(i)} \log\left(\frac{1}{1 + e^{-w^T x^{(i)}}}\right) + (1 - y^{(i)}) \log\left(1 - \frac{1}{1 + e^{-w^T x^{(i)}}}\right) \\
&= -y^{(i)} \log(1 + e^{-w^T x^{(i)}}) - (1 - y^{(i)}) \log(1 + e^{w^T x^{(i)}})
\end{aligned}$$

3.Logistic regression

The derivation process: $\frac{\partial}{\partial w_j}J(w) = \frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})x_j^{(i)}$

$$\frac{\partial}{\partial w_j}J(w) = \frac{\partial}{\partial w_j}(-\frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\log\left(1+e^{-w^T x^{(i)}}\right) - (1-y^{(i)})\log\left(1+e^{w^T x^{(i)}}\right)))$$

$$=-\frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\frac{-x_j^{(i)}e^{-w^T x^{(i)}}}{1+e^{-w^T x^{(i)}}} - (1-y^{(i)})\frac{x_j^{(i)}e^{w^T x^{(i)}}}{1+e^{w^T x^{(i)}}})$$

$$=-\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h(x^{(i)}))x_j^{(i)}$$

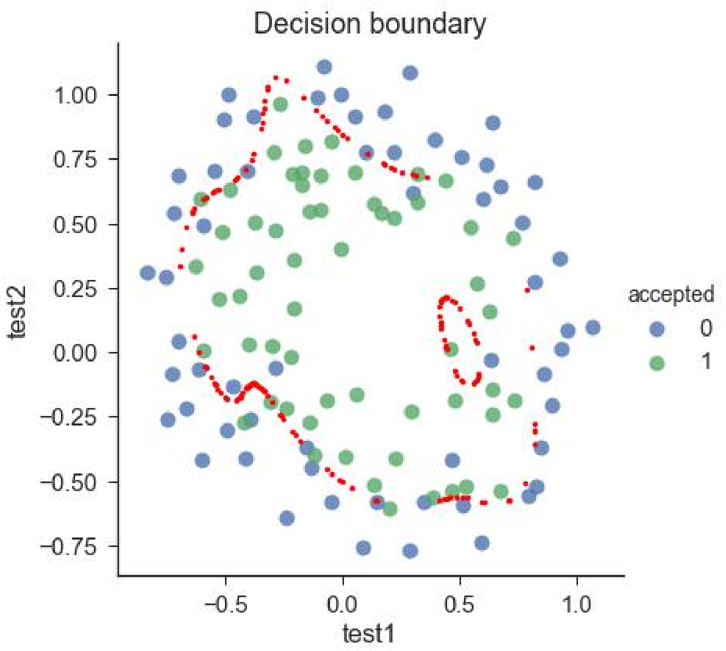
$$=\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})x_j^{(i)}$$

3.Logistic regression

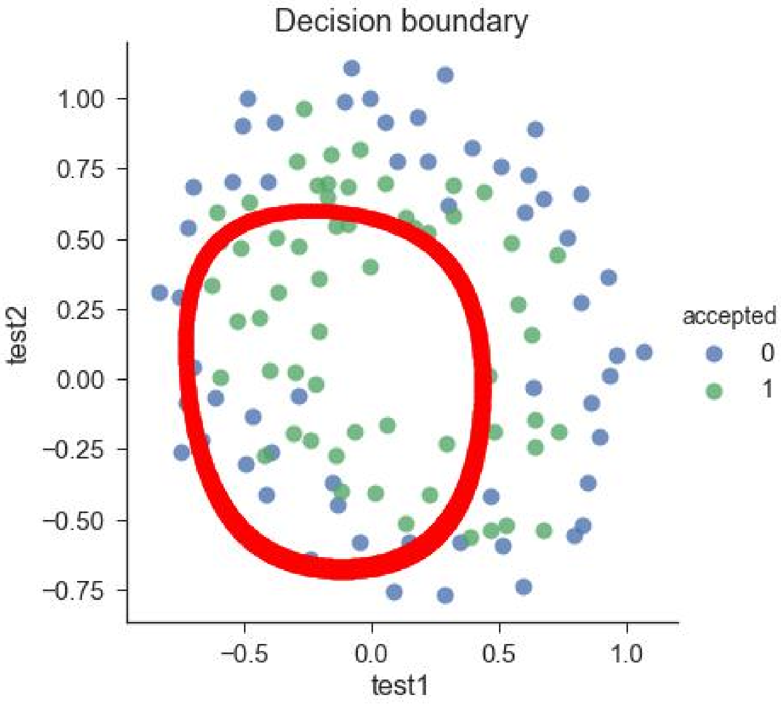
Regularization： to prevent overfitting

regularizer

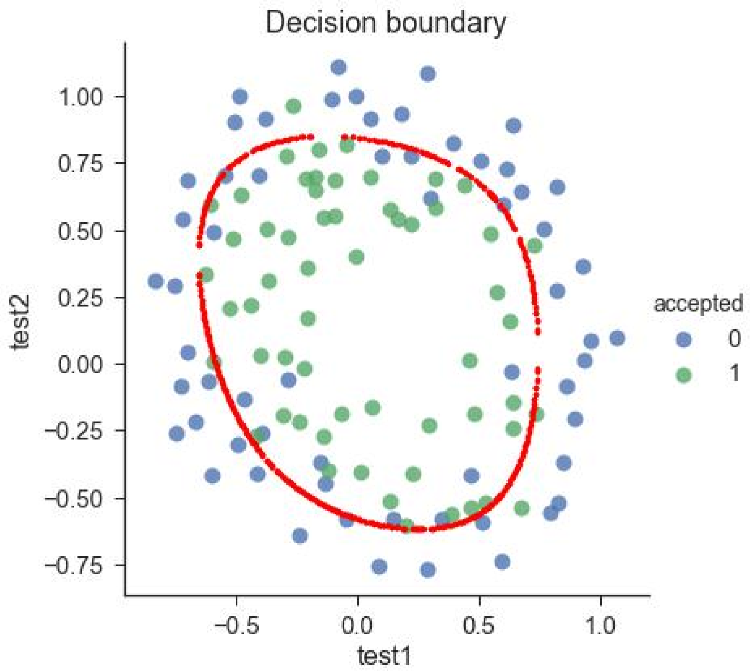
$$J(w)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h\left(x^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)}\right)\right)\right]+\boxed{\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2}$$



no *regularization*



over *regularization*



fit *regularization*

Syllabus

4.Coding

Sigmoid

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

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))
```

cost function

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\log(h(x^{(i)})) + (1-y^{(i)})\log(1-h(x^{(i)})))$$

```
def cost(w, X, y):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))
    return np.sum(first - second) / (len(X))
```

4.Coding

正则化

$$J(w)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h\left(x^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)}\right)\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2$$

```python
def costReg(w, X, y, learningRate):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))
    reg = (learningRate /
           (2 * len(X))) * np.sum(np.power(w[:, 1:w.shape[1]], 2))
    return np.sum(first - second) / len(X) + reg
```

Reference

1. Prof. Andrew Ng. Machine Learning. Stanford University
2. 《统计学习方法》，清华大学出版社，李航著，2019年出版
3. 《机器学习》，清华大学出版社，周志华著，2016年出版
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006

谢 谢！