

Supervised Machine Learning: Regression and Classification

Week 3

Classification with logistic regression

Classification

Question	Answer "y"
Is this email <u>spam</u> ?	no yes
Is the transaction <u>fraudulent</u> ?	no yes
Is the tumor <u>malignant</u> ?	no yes

y can only be one of two values

"binary classification"

class = category

0 1

false true

useful for classification

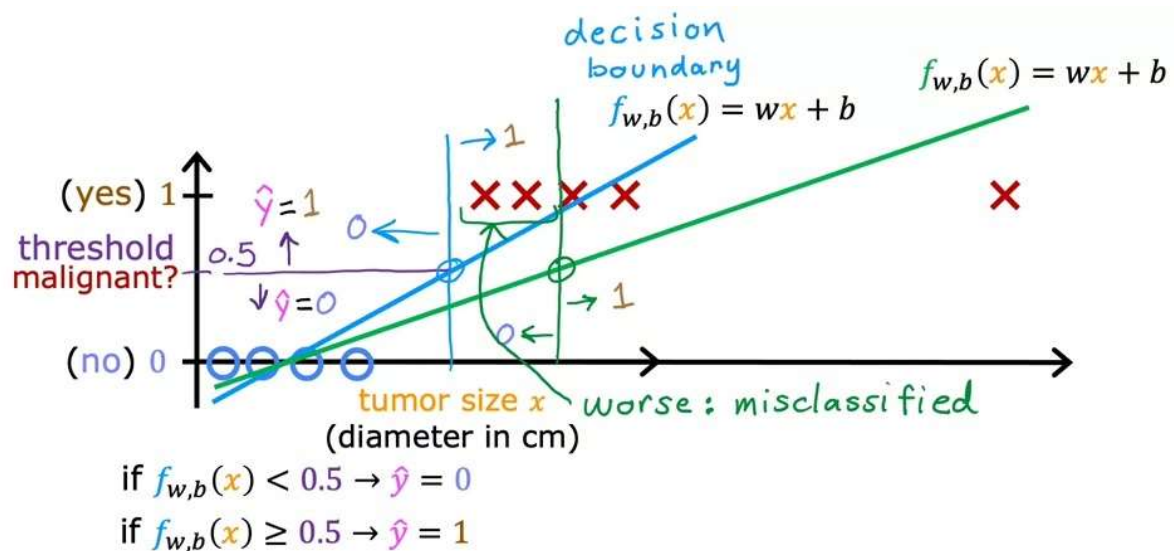
"negative class" \neq "bad" absence

"positive class" \neq "good" presence

Stanford ONLINE

DeepLearning.AI

Andrew Ng

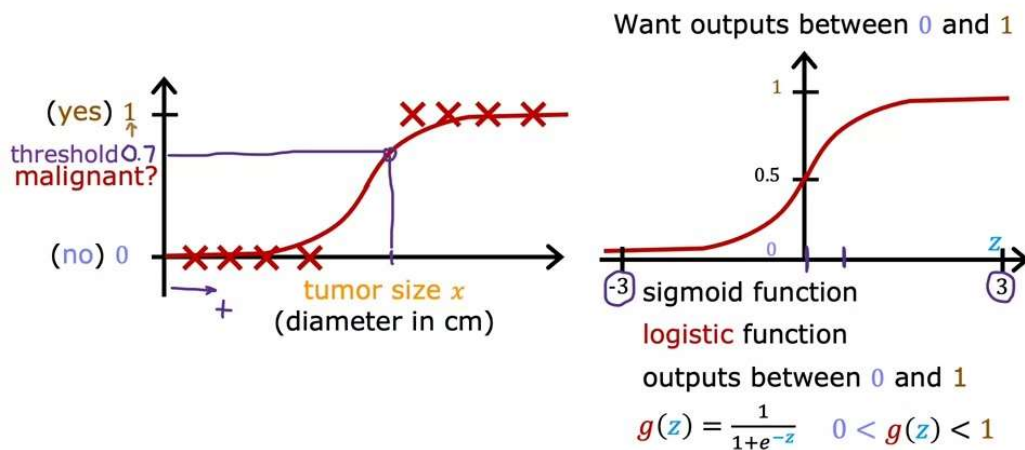


Stanford ONLINE

DeepLearning.AI

Andrew Ng

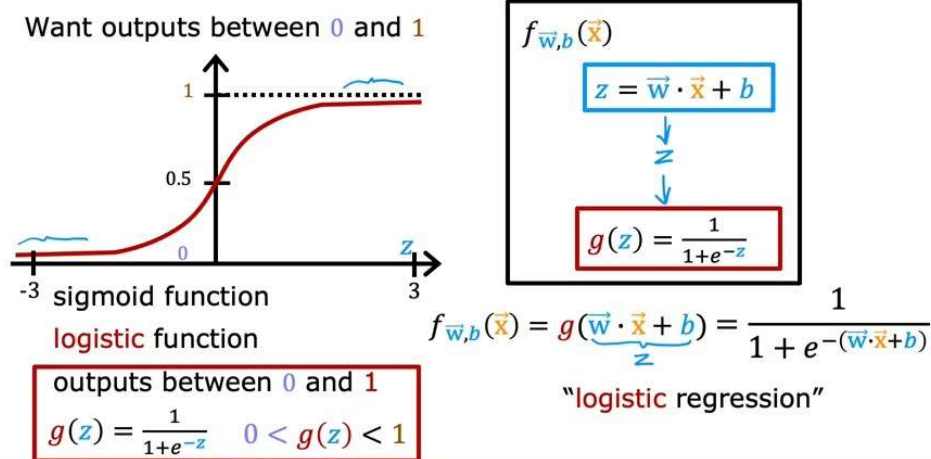
Logistic regression



Stanford ONLINE

DeepLearning.AI

Andrew Ng



Stanford ONLINE

DeepLearning.AI

Andrew Ng

Interpretation of logistic regression output

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w} \cdot \vec{x} + b)}}$$

"probability" that class is 1

Example:

x is "tumor size"
 y is 0 (not malignant)
 or 1 (malignant)

$f_{\vec{w},b}(\vec{x}) = 0.7$
 70% chance that y is 1

$$f_{\vec{w},b}(\vec{x}) = P(y = 1 | \vec{x}; \vec{w}, b)$$

Probability that y is 1,
 given input \vec{x} , parameters \vec{w}, b

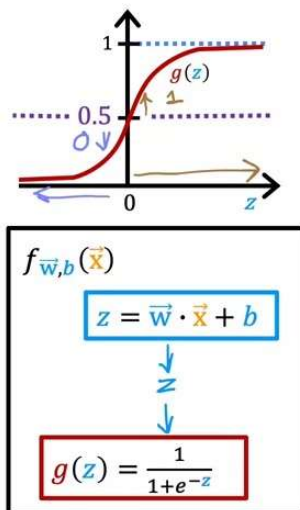
$$P(y = 0) + P(y = 1) = 1$$

Stanford ONLINE

DeepLearning.AI

Andrew Ng

Decision boundary



$$f_{\bar{w},b}(\bar{x}) = g(\underbrace{\bar{w} \cdot \bar{x} + b}_z) = \frac{1}{1 + e^{-z}}$$

$$= P(y = 1 | \bar{x}; \bar{w}, b) \quad 0.7 \quad 0.3$$

0 or 1? threshold

Is $f_{\bar{w},b}(\bar{x}) \geq 0.5$?

Yes: $\hat{y} = 1$

No: $\hat{y} = 0$

When is $f_{\bar{w},b}(\bar{x}) \geq 0.5$?

$$g(z) \geq 0.5$$

$$z \geq 0$$

$$\bar{w} \cdot \bar{x} + b \geq 0$$

$$\hat{y} = 1$$

$$\bar{w} \cdot \bar{x} + b < 0$$

$$\hat{y} = 0$$

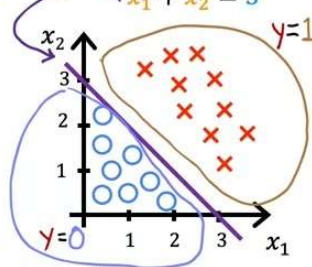
Decision boundary

$$f_{\bar{w},b}(\bar{x}) = g(z) = g(\underbrace{w_1 x_1 + w_2 x_2 + b}_z)$$

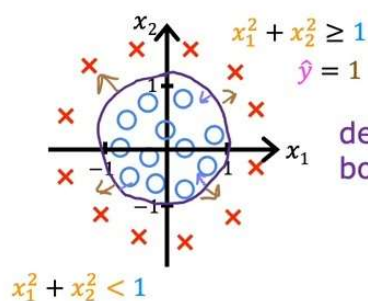
Decision boundary $z = \bar{w} \cdot \bar{x} + b = 0$

$$z = x_1 + x_2 - 3 = 0$$

$$x_1 + x_2 = 3$$



Non-linear decision boundaries



$$f_{\bar{w},b}(\bar{x}) = g(z) = g(\underbrace{w_1 x_1^2 + w_2 x_2^2 + b}_z)$$

$$\text{decision boundary } z = x_1^2 + x_2^2 - 1 = 0$$

$$x_1^2 + x_2^2 = 1$$

$$x_1^2 + x_2^2 < 1$$

Cost function for logistic regression

Training set

	tumor size (cm) x_1	...	patient's age x_n	malignant? y	
$i=1$	10		52	1	$i = 1, \dots, m \leftarrow \text{training examples}$
\vdots	2		73	0	$j = 1, \dots, n \leftarrow \text{features}$
\vdots	5		55	0	target y is 0 or 1
\vdots	12		49	1	$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$
$i=m$	

How to choose $\vec{w} = [w_1 \ w_2 \ \dots \ w_n]$ and b ?

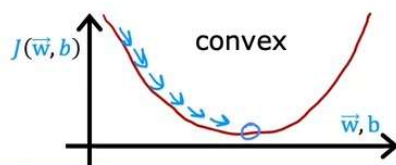
Squared error cost

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})^2$$

loss $L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)})$

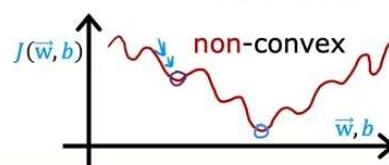
linear regression

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$



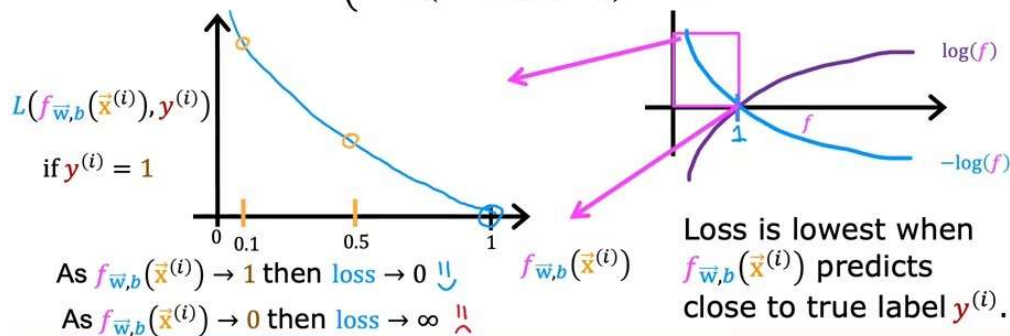
logistic regression

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$



Logistic loss function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$



Simplified Cost Function for Logistic Regression

Simplified loss function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1 - y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

if $y^{(i)} = 1$:

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -1 \log(f(\vec{x}))$$

if $y^{(i)} = 0$:

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = - (1 - 0) \log(1 - f(\vec{x}))$$

Gradient descent for logistic regression

Gradient descent

cost

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))]$$

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b) \quad \frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b) \quad \frac{\partial}{\partial b} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

} simultaneous updates

Gradient descent for logistic regression

repeat { looks like linear regression!

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^m (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \right]$$

} simultaneous updates

Same concepts:

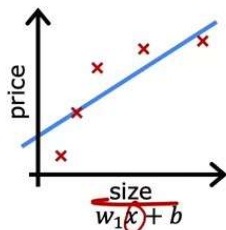
- Monitor gradient descent (learning curve)
- Vectorized implementation
- Feature scaling

Linear regression $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Logistic regression $f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$

Problem of overfitting

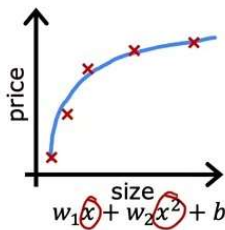
Regression example



underfit

- Does not fit the training set well

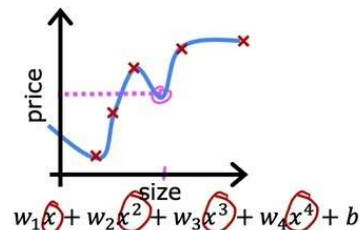
high bias



just right

- Fits training set pretty well

generalization

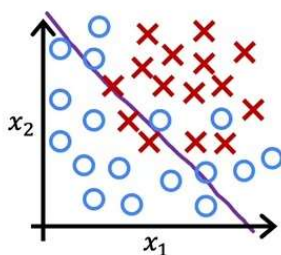


overfit

- Fits the training set extremely well

high variance

Classification

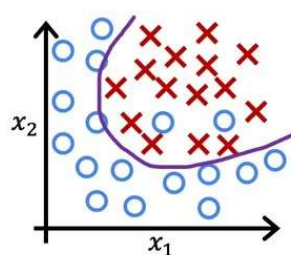


$$z = w_1x_1 + w_2x_2 + b$$

$$f_{\vec{w},b}(\vec{x}) = g(z)$$

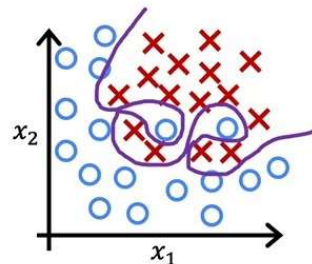
g is the sigmoid function

underfit high bias



$$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$

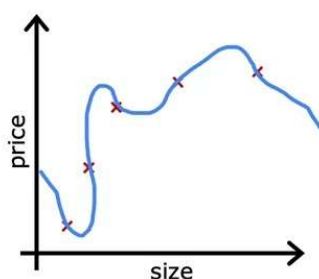
just right



$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + w_6x_1^3x_2 + \dots + b$$

overfit

Collect more training examples

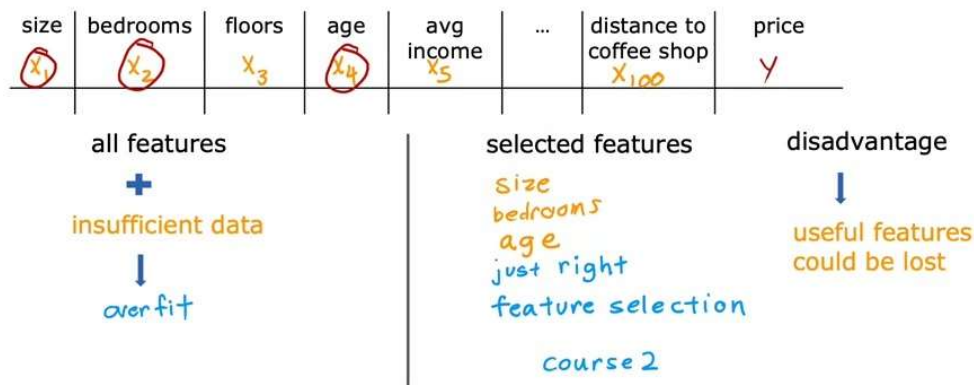


overfit



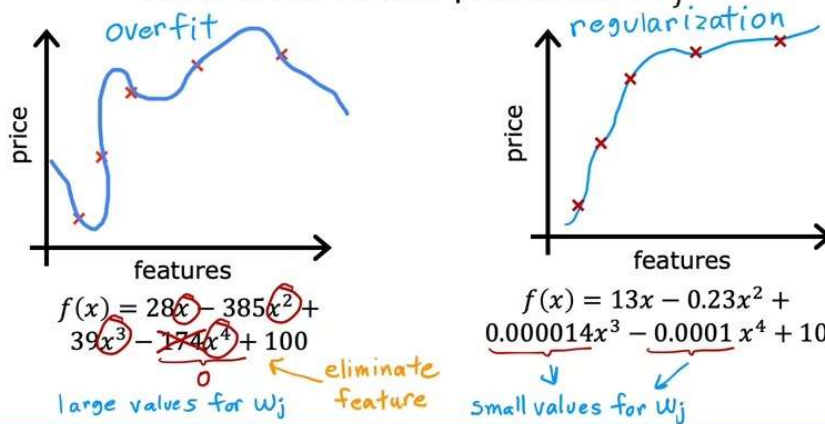
collect more training examples

Select features to include/exclude



Regularization

Reduce the size of parameters w_j



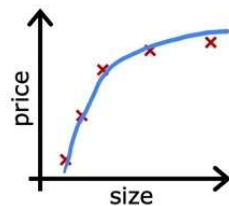
Addressing overfitting

Options

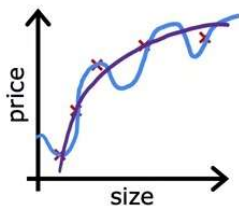
1. Collect more data
2. Select features
 - Feature selection in course 2
3. Reduce size of parameters
 - "Regularization" next videos!

Cost function with regularization

Intuition



$$w_1x + w_2x^2 + b$$



$$w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

≈ 0 ≈ 0

make w_3, w_4 really small (≈ 0)

$$\min_{\vec{w}, b} \frac{1}{2m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 + 1000 \underbrace{w_3^2}_{0.001} + 1000 \underbrace{w_4^2}_{0.002}$$

Stanford ONLINE

DeepLearning.AI

Andrew Ng

Regularization

small values w_1, w_2, \dots, w_n, b

simpler model

less likely to overfit

$$w_3 \approx 0$$

$$w_4 \approx 0$$

size	bedrooms	floors	age	avg income	...	distance to coffee shop	price
x_1	x_2	x_3	x_4	x_5	...	x_{100}	y
$w_1, w_2, \dots, w_{100}, b$							
n features							
$n = 100$							

$$J(\vec{w}, b) = \frac{1}{2m} \left[\sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 + \underbrace{\lambda \sum_{j=1}^n w_j^2}_{\text{regularization term}} + \underbrace{\lambda b^2}_{\text{can include or exclude } b} \right]$$

regularization parameter $\lambda > 0$

Stanford ONLINE

DeepLearning.AI

Andrew Ng

Regularization

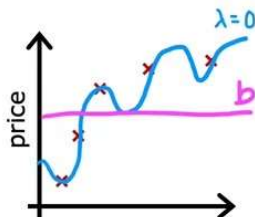
$$\min_{\vec{w}, b} J(\vec{w}, b) = \min_{\vec{w}, b} \left[\underbrace{\frac{1}{2m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2}_{\text{mean squared error}} + \underbrace{\frac{\lambda}{2m} \sum_{j=1}^n w_j^2}_{\text{regularization term}} \right]$$

fit data \rightarrow λ balances both goals \rightarrow Keep w_j small

choose $\lambda = 10^{10}$

$$f_{\vec{w}, b}(\vec{x}) = \underbrace{w_1x}_{\approx 0} + \underbrace{w_2x^2}_{\approx 0} + \underbrace{w_3x^3}_{\approx 0} + \underbrace{w_4x^4}_{\approx 0} + b$$

$f(x) = b$ choose λ



Stanford ONLINE

DeepLearning.AI

Andrew Ng

Regularized linear regression

Regularized linear regression

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

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$j=1, \dots, n$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

} simultaneous update

$$= \frac{1}{m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} w_j$$

$$= \frac{1}{m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})$$

don't have to
regularize b

Implementing gradient descent

repeat {

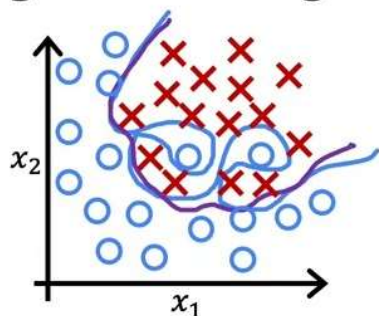
$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left[(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right] + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})$$

} simultaneous update

Regularized logistic regression

Regularized logistic regression



$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + \dots + b$$

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-z}}$$

Cost function

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

$\min_{\vec{w}, b} J(\vec{w}, b) \rightarrow w_j \downarrow$

Regularized logistic regression

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

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$j = 1 \dots n$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

}

Looks same as for linear regression!

$$= \frac{1}{m} \sum_{i=1}^m \left[(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right] + \frac{\lambda}{m} w_j$$

logistic regression

don't have to regularize