Review of Linear Regression

Linear models

Problem: $\{(x_i, y_i)\}$.

Given x, predict $\hat{y}$.

Linear models

x: **explanatory** or **predictor** variable. Or the **signal**.

y: **response** variable.

For some reason, we believe a linear model is a good idea.

Residuals

Residuals (model error):

- What's left over
- What the model doesn't explain

$$\text{data} = \text{fit} + \text{residual}$$

Residuals

Residuals (model error):

- What's left over
- What the model doesn't explain

$$y_i = \hat{y}_i + e_i$$

Residuals

Residuals (model error):

- What's left over
- What the model doesn't explain

$$e_i = y_i - \hat{y}_i$$

Residuals

Residuals (model error):

- What's left over
- What the model doesn't explain

Goal: small residuals.

$$\sum e_i^2$$

Hypothesis (Model)

$$y = a + bx$$

Hypothesis (Model)

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

Cost Function

Also called the “loss function”.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (e_i)^2$$

Cost Function

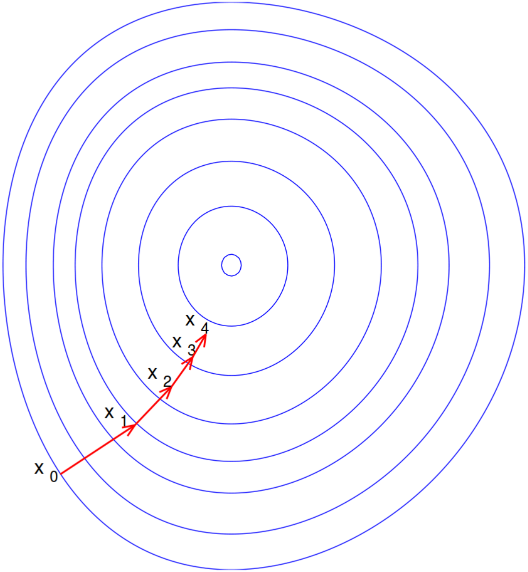
Also called the "loss function".

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

Cost Function

Also called the “loss function”.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$


x 0
x 1
x 2
x 3
x 4

Gradient Descent

$$\begin{cases} \theta_0 & \leftarrow \theta_0 - \alpha \dfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\ \theta_1 & \leftarrow \theta_1 - \alpha \dfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \end{cases}$$

Gradient Descent

$$
\begin{cases}
\theta_0 & \leftarrow \theta_0 - \alpha \dfrac{\partial}{\partial \theta_0} \left(\dfrac{1}{2m} \displaystyle\sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 \right) \\
\theta_1 & \leftarrow \theta_1 - \alpha \dfrac{\partial}{\partial \theta_1} \left(\dfrac{1}{2m} \displaystyle\sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 \right)
\end{cases}
$$

Gradient Descent

$$
\begin{cases}
\theta_0 & \leftarrow \theta_0 - \alpha \dfrac{\partial}{\partial \theta_0} \left(\dfrac{1}{2m} \displaystyle\sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i)^2 \right) \\
\theta_1 & \leftarrow \theta_1 - \alpha \dfrac{\partial}{\partial \theta_1} \left(\dfrac{1}{2m} \displaystyle\sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i)^2 \right)
\end{cases}
$$

Gradient Descent

$$\begin{cases} \theta_0 & \leftarrow \theta_0 - \dfrac{\alpha}{m} \displaystyle\sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i)\, 1 \\ \theta_1 & \leftarrow \theta_1 - \dfrac{\alpha}{m} \displaystyle\sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i) - y_i)\, x_i \end{cases}$$

Gradient Descent

$$\begin{cases} \theta_0 & \leftarrow \theta_0 - \dfrac{\alpha}{m} \displaystyle\sum_{i=1}^{m} (h_\theta(x_i) - y_i) \\ \theta_1 & \leftarrow \theta_1 - \dfrac{\alpha}{m} \displaystyle\sum_{i=1}^{m} (h_\theta(x_i) - y_i)\, x_i \end{cases}$$

Logistic regression

Linear regression

- Inputs are continuous or discrete
- Continuous output
- Normal residues
- Predict $\hat{y}$ for x given $\{(x_i, y_i)\}$

Logistic regression

- Inputs are continuous or discrete
- Binary output
- Classification

Logistic regression

- Have: continuous and discrete inputs
- Want: class (0 or 1)

Logistic regression: motivation

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

$$h_\theta(x) = .75 \iff \text{event has 75\% of being true}$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

$$h_\theta(x) = \Pr(y = 1 \mid x; \theta) = 0.75$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

So this must be true:

$$\Pr(y = 0 \mid x; \theta) + \Pr(y = 1 \mid x; \theta) = 1$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

$$\text{Set } y = 1 \iff h_\theta(x) = \Pr(y = 1 \mid x; \theta) > \tfrac{1}{2}$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

Let

$$Pr(y = 0 \mid x; \theta) = h_\theta(x)$$
$$Pr(y = 1 \mid x; \theta) = 1 - h_\theta(x)$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

Let

$$Pr(y = 0 \mid x; \theta) = h_\theta(x)$$
$$Pr(y = 1 \mid x; \theta) = 1 - h_\theta(x)$$

Then

$$Pr(y = 0 \mid x; \theta) + \Pr(y = 1 \mid x; \theta) = 1$$

Probabilistic inspiration

The probabilities are motivations: this doesn't really behave like a probability.

Math review:

- $z = (\theta^T x)$
- $\theta^T x \geqslant 0 \iff h_\theta \geqslant 0.5$
- $\theta^T x \geqslant 0 \iff$ predict $y = 1$

Logistic (sigmoid) function

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

Logistic (sigmoid) function

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

Exercise: plot this

Logistic (sigmoid) function

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

Let

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

Logistic (sigmoid) function

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

Then

$$\sigma(z) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

Cost function in logistic regression

In linear regression, we had

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (e_i)^2$$

Cost function in logistic regression

In linear regression, we had

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

Cost function in logistic regression

In linear regression, we had

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$$

Cost function in logistic regression

In linear regression, we had

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

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

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

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

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

$$\frac{dg}{dz} = \frac{-1}{(1 - e^{-z})^2} \cdot \left(e^{-z}\right)$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

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

$$\frac{dg}{dz} = \frac{-1}{(1 - e^{-z})^2} \cdot \left(e^{-z}\right)$$

$$= \frac{e^{-z}}{(1 - e^{-z})^2}$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\begin{aligned}
\frac{dg}{dz} &= \frac{-1}{\left(1 - e^{-z}\right)^2} \cdot \left(e^{-z}\right) \\
&= \frac{e^{-z}}{\left(1 - e^{-z}\right)^2} \\
&= \frac{e^{-z}}{\left(1 - e^{-z}\right)\left(1 - e^{-z}\right)}
\end{aligned}$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\begin{aligned}
\frac{dg}{dz} &= \frac{e^{-z}}{(1-e^{-z})^2} \\
&= \frac{e^{-z}}{(1-e^{-z})\,(1-e^{-z})} \\
&= \left(\frac{1}{1-e^{-z}}\right)\left(\frac{e^{-z}}{1-e^{-z}}\right)
\end{aligned}$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$
\begin{aligned}
\frac{dg}{dz} &= \frac{e^{-z}}{\left(1 - e^{-z}\right)\left(1 - e^{-z}\right)} \\
&= \left(\frac{1}{1 - e^{-z}}\right)\left(\frac{e^{-z}}{1 - e^{-z}}\right) \\
&= \left(\frac{1}{1 - e^{-z}}\right)\left(\frac{1 - \left(1 - e^{-z}\right)}{1 - e^{-z}}\right)
\end{aligned}
$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\begin{aligned}\frac{dg}{dz} &= \left(\frac{1}{1-e^{-z}}\right)\left(\frac{e^{-z}}{1-e^{-z}}\right) \\ &= \left(\frac{1}{1-e^{-z}}\right)\left(\frac{1-(1-e^{-z})}{1-e^{-z}}\right) \\ &= \left(\frac{1}{1-e^{-z}}\right)\left(1-\frac{1}{1-e^{-z}}\right)\end{aligned}$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

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

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\frac{dg}{dz} = \left(\frac{1}{1 - e^{-z}}\right)\left(1 - \frac{1}{1 - e^{-z}}\right)$$

$$= g(z)(1 - g(z))$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\frac{dg}{dz} = g(z)(1 - g(z))$$

Cost function in logistic regression

To do gradient descent, we need the derivative of the cost function.

$$\frac{dg}{dz} = g(z)(1 - g(z))$$

And for today we'll stop there, coming back to this next week.

null hypothesis

true positive, true negative

false positive, false negative

type I error
(incorrect rejection of null hypothesis)

type II error
(failure to reject null hypothesis)

sensitivity

100% sensitivity = no false negatives

specificity

100% specificity = no false positives

Precision

$$P = \frac{TP}{TP + FP}$$

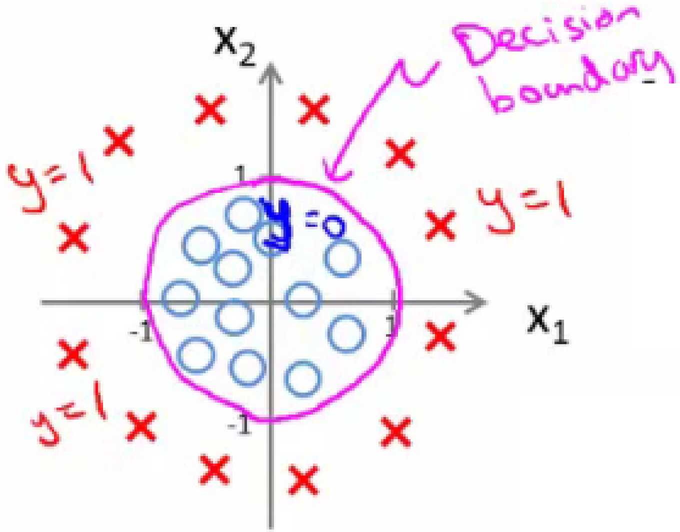
Recall

$$R = \frac{TP}{TP + FN}$$

F1 score

$$F1 = \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Non-linear decision boundaries



Non-linear decision boundaries

OvA = OvR

OvO

Non-linear decision boundaries

One vs All = One vs Rest

One vs One

Scikit Learn

```
>>> import numpy as np
>>> from sklearn.linear_model import LinearRegression
>>> X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
>>> # y = 1 * x_0 + 2 * x_1 + 3
>>> y = np.dot(X, np.array([1, 2])) + 3
>>> reg = LinearRegression().fit(X, y)
>>> reg.score(X, y)
1.0
>>> reg.coef_
array([1., 2.])
>>> reg.intercept_
3.0000...
>>> reg.predict(np.array([[3, 5]]))
array([16.])
```

Scikit Learn

```
>>> import numpy as np
>>> from sklearn.linear_model import LogisticRegression
>>> X = np.array([[1, 0], [0, 1], [3, 2], [2, 3]])
>>> y = np.array([0, 0, 1, 1])
>>> reg = LogisticRegression(solver='lbfgs').fit(X, y)
>>> reg.score(X, y)
1.0
>>> reg.coef_
array([1., 2.])
>>> reg.intercept_
3.0000...
>>> reg.predict(np.array([[3, 5]]))
array([1])
```