Machine learning diagnostic

Debugging a learning algorithm

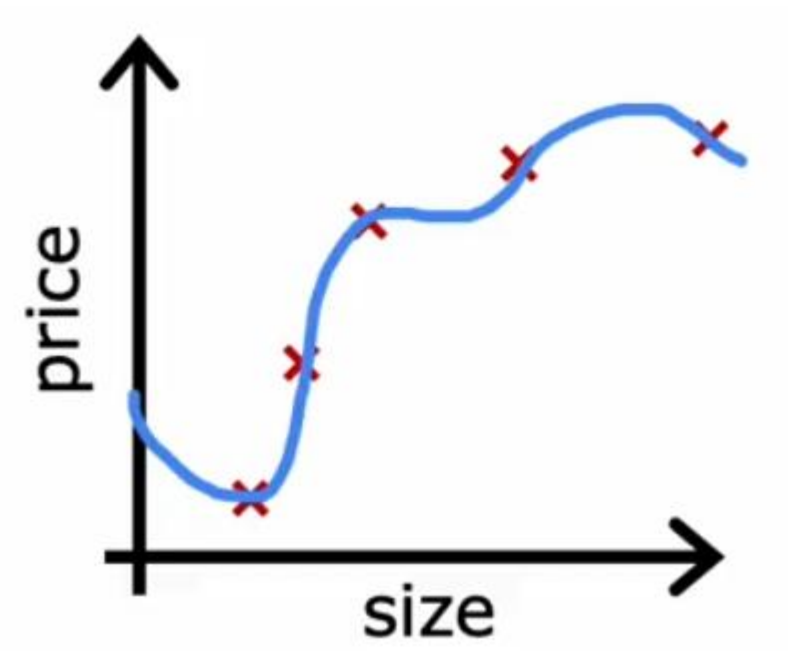
- Regularized linear regression

- Large errors in prediction
- Solutions

1. Get more training examples
2. Try smaller set of features
3. Try getting additional features
4. Try adding polynomial features
5. Try decreasing λ
6. Try increasing λ

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

Evaluating model



$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + \cdots + w_n x^n + b$$

Issue: Model fits the training data well but will fail to generalize to new examples not in the training set

x_1= size in feet^2
x_2= no. of bedrooms
x_3= no. of floors
x_4= age of home in years

Evaluating model (Training/Test set)

Dataset:

size	price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

m_{train} = no. training examples

$$\left(x^{(1)}, y^{(1)}\right)$$
$$\left(x^{(2)}, y^{(2)}\right)$$
$$\vdots$$
$$\left(x^{(m_{train})}, y^{(m_{train})}\right)$$

m_{test} = no. test examples

$$\left(x_{test}^{(1)}, y_{test}^{(1)}\right)$$
$$\vdots$$
$$\left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)$$

Train/test procedure for linear regression

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

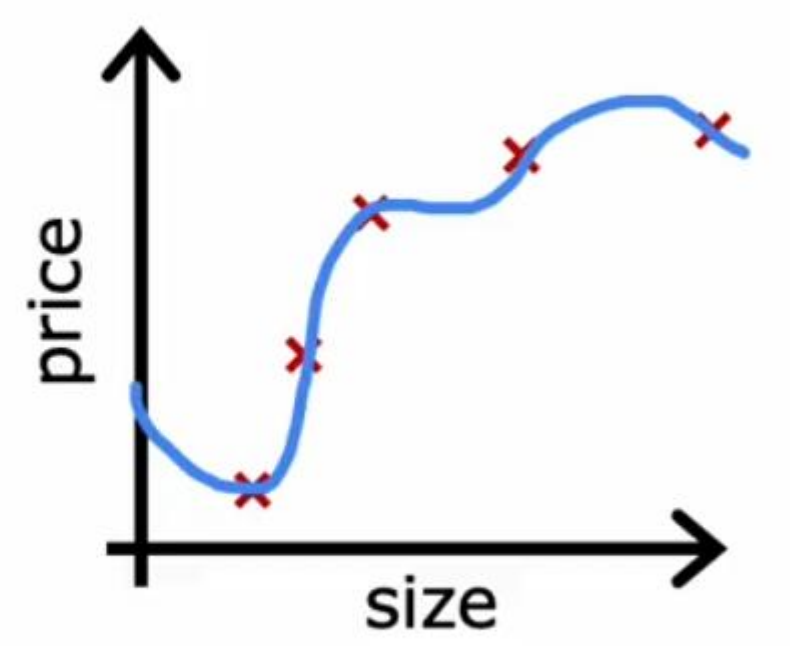
Compute test error:

$$J_{test}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{test}} \left[\sum_{i=1}^{m_{test}} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right) - y_{test}^{(i)} \right)^2 \right]$$

Compute training error:

$$J_{train}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{train}} \left[\sum_{i=1}^{m_{train}} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{train}^{(i)}\right) - y_{train}^{(i)} \right)^2 \right]$$

Train/test procedure for linear regression



Train/test procedure for logistic regression

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

Compute test error:

$$J_{test}(\vec{\mathrm{w}},b) = -\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\left[y_{test}^{(i)}\log\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right)\right) + \left(1-y_{test}^{(i)}\right)\log\left(1-f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right)\right)\right]$$

Compute train error:

$$J_{train}(\vec{\mathrm{w}},b) = -\frac{1}{m_{train}}\sum_{i=1}^{m_{train}}\left[y_{train}^{(i)}\log\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{train}^{(i)}\right)\right) + \left(1-y_{train}^{(i)}\right)\log\left(1-f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{train}^{(i)}\right)\right)\right]$$

Train/test procedure for logistic regression

fraction of the test set and the fraction of the train set that the algorithm has misclassified.

$$\hat{y} = \begin{cases} 1 \text{ if } f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}^{(i)}\right) \geq 0.5 \\ 0 \text{ if } f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}^{(i)}\right) < 0.5 \end{cases}$$

count $\hat{y} \neq y$

$J_{test}(\vec{\mathrm{w}}, b)$ is the fraction of the test set that has been misclassified.

$J_{train}(\vec{\mathrm{w}}, b)$ is the fraction of the train set that has been misclassified.

Model selection

1. $f_{\vec{w},b}(\vec{x}) = w_1x + b$
2. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + b$
3. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + b$

   $\vdots$

10. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + \cdots + w_{10}x^{10} + b$

How well does the model perform? Report test set error $J_{test}(w^{<5>}, b^{<5>})$?

The problem: $J_{test}(w^{<5>}, b^{<5>})$ is likely to be an optimistic estimate of generalization error (ie. $J_{test}(w^{<5>}, b^{<5>}) < \text{generalization error}$). Because an extra parameter d (degree of polynomial) was chosen using the test set.

w, b are overly optimistic estimate of generalization error on training data.

Training/Cross validation/Test set

Dataset:

size	price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

$$\left(x^{(1)}, y^{(1)}\right)$$
$$\vdots$$
$$\left(x^{(m_{train})}, y^{(m_{train})}\right)$$

$$\left(x_{cv}^{(1)}, y_{cv}^{(1)}\right)$$
$$\vdots$$
$$\left(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})}\right)$$

$$\left(x_{test}^{(1)}, y_{test}^{(1)}\right)$$
$$\vdots$$
$$\left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)$$

Training/Cross validation/Test set

Training error: $J_{train}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{train}}\left[\sum_{i=1}^{m_{train}} \left(f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}^{(i)}) - y^{(i)}\right)^2\right]$

Cross validation error: $J_{cv}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{cv}}\left[\sum_{i=1}^{m_{cv}} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{cv}^{(i)}\right) - y_{cv}^{(i)}\right)^2\right]$ (validation error, dev error)

Test error: $J_{test}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}} \left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2\right]$

Model selection

1. $f_{\vec{w},b}(\vec{x}) = w_1x + b$
2. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + b$
3. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + b$

   $\vdots$

10. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + \cdots + w_{10}x^{10} + b$

How well does the model perform? Report test set error $J_{test}(w^{<5>}, b^{<5>})$?
The problem: $J_{test}(w^{<5>}, b^{<5>})$ is likely to be an optimistic estimate of generalization error (ie. $J_{test}(w^{<5>}, b^{<5>}) < \text{generalization error}$). Because an extra parameter d (degree of polynomial) was chosen using the test set.

w, b are overly optimistic estimate of generalization error on training data.

R^2 score (R-squared)

- Metric that measures how well a regression model explains the variance in the target variable
- Coefficient of determination

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where:

- $SS_{res} = \sum(y_{true} - y_{pred})^2$ (Sum of Squared Residuals)
- $SS_{tot} = \sum(y_{true} - \bar{y})^2$ (Total Sum of Squares)
- y_{true} = Actual values
- y_{pred} = Predicted values
- $\bar{y}$ = Mean of actual values

Interpretation:

- $R^2 = 1$ → Perfect fit (Model explains 100% of variance)
- $R^2 = 0$ → Model explains no variance (Same as predicting the mean)
- $R^2 < 0$ → Model performs worse than the mean prediction

Insurance Dataset

age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.4706
32	male	28.88	0	no	northwest	3866.8552

Steps to Follow:

1. Load the dataset.
2. Perform exploratory data analysis (EDA).
3. Train a baseline **Linear Regression** model.
4. Apply **Ridge Regression** (L2) to see how it reduces overfitting.
5. Apply **Lasso Regression** (L1) to observe feature selection effects.
6. Compare model performance using **R^2 score and Mean Squared Error (MSE).**