Regularisation

Intelligent Systems and Control

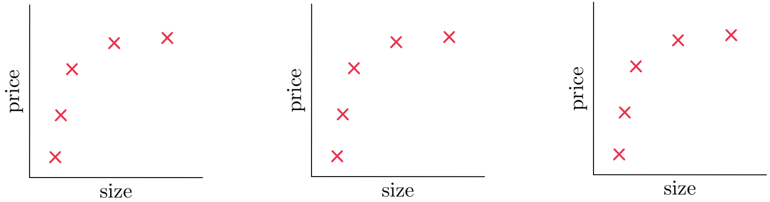
2019

Sepehr Maleki

University of Lincoln
School of Engineering

The Problem of Overfitting/Underfitting

Example: Linear Regression



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

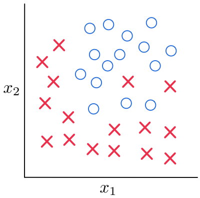
Overfitting: If we have too many features, the learned model may fit training set too well (cost ≈ 0), but fail to generalise to new examples.

Underfitting: If we have very few features, such that the learned model does not fit the data well enough (large cost).

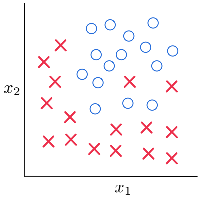
(S. Maleki 2019)

The Problem of Overfitting/Underfitting

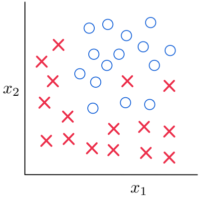
Example: Logistic Regression



$\sigma(w_2x_2 + w_1x_1 + b)$



$$\sigma(w_5x_1x_2 + w_4x_2^2 + w_3x_1^2 \\ + w_2x_2 + w_1x_1 + b)$$



$$\sigma(w_6x_1^3x_2 + w_5x_1^2x_2^3 + w_4x_1^2x_2^2 \\ + w_3x_1^2x_2 + w_2x_1^2 + w_1x_1 + b)$$

(S. Maleki 2019)

Addressing The Overfitting Problem

I. Reduce number of features:

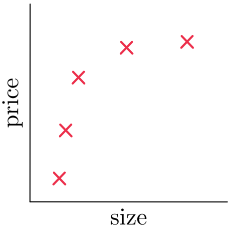
- Manually select which features to keep.
- Model selection.

II. Regularisation:

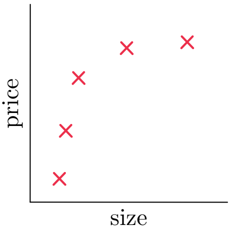
- Keep all the features but reduce magnitude/values of parameters w_j.
- Works well when we have a lot of features, each of which contributes a bit to predicting y.

(S. Maleki 2019)

Introducing a Penalty Term



$w_2x^2 + w_1x + b$



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

Suppose we denote the cost function we used to minimise by $C(\mathbf{w})$. We add a Penalty term, $P(\mathbf{w})$, to this function to penalise the parameters. So now our objective is to minimise:

$$J(\mathbf{w}) = C(\mathbf{w}) + P(\mathbf{w}) \; .$$

(S. Maleki 2019)

The Idea of Regularisation

Regularisation favours small values for parameters, $w_1, w_2, ..., w_d$, which results in simpler models that are less prone to overfitting.

We define the penalty term as follows:

$$P(\mathbf{w}) = \lambda \sum_{i=1}^{d} w_j^2 \ ,$$

where λ is the regularisation parameter. Therefore, the overall cost function becomes:

$$J(\mathbf{w}) = C(\mathbf{w}) + \lambda \sum_{i=1}^{d} w_j^2 \ .$$

(S. Maleki 2019)

Regularised Linear Regression

We defined the cost function for linear regression by:

$$C(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\Big(f(\mathbf{x}_i) - y_i\Big)^2 .$$

The cost function with the regularisation term is therefore given by:

$$J(\mathbf{w}) = \frac{1}{2}\Bigg[\sum_{i=1}^{n}\Big(f(\mathbf{x}_i) - y_i\Big)^2 + \lambda\sum_{i=1}^{d} w_j^2\Bigg] .$$

(S. Maleki 2019)

Regularised Logistic Regression

We defined the logistic regression cost function by:

$$C(\mathbf{w}) = -\frac{1}{n}\left(\sum_{i=1}^{n} c_i \log f(\mathbf{x}_i) + (1-c_i)\log(1-f(\mathbf{x}_i))\right) \,,$$

which with the regularisation term becomes:

$$J(\mathbf{w}) = -\frac{1}{n}\left(\sum_{i=1}^{n} c_i \log f(\mathbf{x}_i) + (1-c_i)\log(1-f(\mathbf{x}_i))\right) + \lambda \sum_{i=1}^{d} w_j^2 \;.$$

(S. Maleki 2019)

Gradient Descent For The New Cost Function

To obtain the best parameters via the gradient descent algorithm, we repeatedly performed:

$$w_j = w_j - \alpha \sum_{i=1}^{n} \Big(f(\mathbf{x}_i) - y_i \Big) x_i^{(j)} \ .$$

With regularisation, the new updating rule becomes:

$$w_j = w_j - \alpha \sum_{i=1}^{n} \Big(f(\mathbf{x}_i) - y_i \Big) x_i^{(j)} - \frac{\lambda}{n} w_j \ .$$

(S. Maleki 2019)