Machine Learning W7 Tutorial

COMP30027 | Sandy Luo

Overview

Linear Regression

Concept, code

Gradient Descent

Theory, code

Model Interpretation

Hyperparameters etc.

Linear Regression

Q1:

What is Linear Regression? In what circumstances is it a good choice of model, and in what circumstances it is a poor choice?

Linear Regression

- Captures a linear relationship b/w:
  a. Outcome variable (y)
     - Response variable, dependent variable, label
  b. ≥ 1 predictors (x_1,...,x_D)
     - Independent variable, explanatory variable, feature
- At its most basic, the relationship can be expressed as a line:

$$y = f(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_D x_D = \beta \cdot x$$
$$\text{where } \mathbf{x} = [x_0, x_1, \ldots, x_D], x_0 = 1$$

Q1:

What is Linear Regression? In what circumstances is it a good choice of model, and in what circumstances it is a poor choice?

- Linear model to predict target values by finding a weight for each attribute
  - Therefore, each prediction is a point on a line (hyperplane)
- Tune weights via gradient descent → minimise error of predictions
  - e.g. using MSE (convex), choice depends on amount of noise

- Good choice when (a) want to predict value of target that is dependent on values of independent variables, and (b) underlying relationship is linear

Q2:

What is gradient descent? How is it used in machine learning?

Q2:

What is gradient descent? How is it used in machine learning?

- Iterative optimisation algorithm → minimise the cost/error function
    - Iteratively adjusting the model's parameters in the opposite direction of the gradient of the cost function, in order to minimise the error or cost.

- Useful when no closed-form solution available for finding optimal parameters, as it can iteratively update the parameters until convergence.
    - Popular choice for ML tasks such as linear and logistic regression, neural networks, and deep learning, where the goal = find weights that minimise the error or cost function over a training dataset.

Q3:

Recall that the update rule for Gradient Descent with respect to Mean Squared Error (MSE) is as follows:

$$\beta_k^{t+1} = \beta_k^t + \frac{2\alpha}{N} \sum_{i=1}^{N} x_{ik}(y_i - \hat{y}_i^t)$$

Suppose we wish to fit a linear regression model to predict y from x given the following instances:

x	y
1	1
2	2
2	3

Intialise the model parameters to 0, so the initial model is $y = 0 + 0x$. Set the learning rate to $\alpha = 0.15$ and fit the model using gradient descent.

- How many weights do we have to learn?

Q3: First Iteration

$$\beta_k^{t+1} = \beta_k^t + \frac{2\alpha}{N}\sum_{i=1}^{N} x_{ik}(y_i - \hat{y}_i^t)$$

x	y
1	1
2	2
2	3

- Initial parameters at t=0: \beta = <0,0>, \alpha = 0.15
- Parameters update according to:

$$\beta_k^{(1)} = \beta_k^{(0)} + \frac{2\alpha}{N}\sum_{i=1}^{N} x_{ik}(y_i - \hat{y}_i)$$

1. Evaluate (y_i - \hat{y_i})

$$\hat{y}_1 = 0 + 0(1) = 0, y_1 = 1$$

$$\hat{y}_2 = 0 + 0(2) = 0, y_2 = 2$$

$$\hat{y}_3 = 0 + 0(2) = 0, y_3 = 3$$

2. Update each parameter according to function

$$\beta_0^{(1)} = 0 + \frac{2 * 0.15}{3}(1(1-0) + 1(2-0) + 1(3-0)) = 0.6$$

$$\beta_1^{(1)} = 0 + \frac{2 * 0.15}{3}(1(1-0) + 2(2-0) + 2(3-0)) = 1.1$$

→ y = 0.6 + 1.1*x

Q3: Second Iteration

$$\beta_k^{t+1} = \beta_k^t + \frac{2\alpha}{N}\sum_{i=1}^{N} x_{ik}(y_i - \hat{y}_i^t)$$

x	y
1	1
2	2
2	3

- Initial parameters at t=1: \beta = <0.6, 1.1>, \alpha = 0.1

1. Evaluate (y_i - \hat{y_i})

$$\hat{y}_1 = 0.6 + 1.1(1) = 1.7, y_1 = 1$$

$$\hat{y}_2 = 0.6 + 1.1(2) = 2.8, y_2 = 2$$

$$\hat{y}_3 = 0.6 + 1.1(2) = 2.8, y_3 = 3$$

2. Update each parameter according to function

→ y = 0.47 + 0.91*x

$$\beta_0^{(2)} = 0.6 + \frac{2 * 0.15}{3}(1(1 - 1.7) + 1(2 - 2.8) + 1(3 - 2.8)) = 0.47$$

$$\beta_1^{(2)} = 1.1 + \frac{2 * 0.15}{3}(1(1 - 1.7) + 2(2 - 2.8) + 2(3 - 2.8)) = 0.91$$

and continue...

Model Interpretation

Q5:

What are some examples of hyperparameters in the following algorithms?

1. Naive Bayes

2. Decision tree

3. K-NN

Q5:

What are some examples of hyperparameters in the following algorithms?

1. Naive Bayes choice of smoothing function

2. Decision tree stopping criterion (e.g. minimum IG for split), max depth, minimum number of instances for pruning

3. K-NN

k, distance measure, weighting function

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

1. Would you trust this model to identify patients with the disease, based on this accuracy result? Why or why not?

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

1. Would you trust this model to identify patients with the disease, based on this accuracy result? Why or why not?

Accuracy doesn’t give information about predicted class distributions. Given that this is a very imbalanced case, high accuracy could simply be majority voting and ignoring the minority class, which is bad.

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

2. What type of error is most important in this task?

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

2. What type of error is most important in this task?

- False negatives (predicted as negative, but is actually positive)
- Severe consequences → priority to minimise this, even if sacrifice other criteria

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

3. Name at least one appropriate evaluation metric that you would choose to evaluate your model.

Q5:

You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are known to have the disease. You achieve 96% classification accuracy.

3. Name at least one appropriate evaluation metric that you would choose to evaluate your model.

- Need to measure False negatives (predicted as negative, but is actually positive)
- Recall (TP/TP+FN) directly measures ability to minimise FN
    - High = effective in detecting positive cases
    - Low = Missing positive cases, need further refinement