Scalable Machine Learning and Deep Learning - Review Questions 1

Anna Martignano, Daniele Montesi

May 4, 2020

1. **Which of the following is/are true about Normal Equation?**
(a) We don't have to choose the learning rate. ✓
(b) It becomes slow when number of features is very large. ✓
(c) No need to iterate. ✓

2. **The following graph, Figure 1, represents a regression line predicting y from x. The values on the graph shows the residuals for each predictions value, i.e., $\hat{\mathbf{y}}$ y. Calculate the squared error of the prediction.**

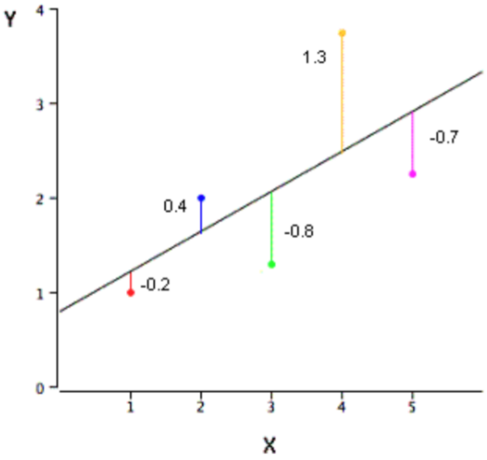


Figure 1: Regression Line

The squared error, also called Residual Sum of Squares (RSS), is calculated by doing the summation of the squared of residuals as written in the following formula:

$$RSS = \sum_{i=1}^{n} (y - \hat{y})^2 = 3.02$$

Dividing RSS by the number of predictions, it is possible to obtain the Mean Squared Error (MSE) which is the average squared difference between the estimated values and the actual value.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y})^2 = 0.604$$

3. **How does number of observations influence overfitting? Choose the correct answer(s).**
   (a) In case of fewer observations, it is easy to overfit the data. ✓
   (b) In case of fewer observations, it is hard to overfit the data.
   (c) In case of more observations, it is easy to overfit the data.
   (d) In case of more observations, it is hard to overfit the data. ✓

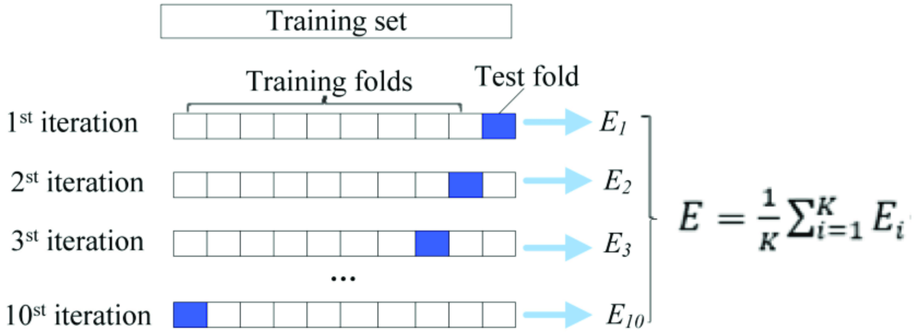
4. **How many coefficients do you need to estimate in a simple linear regression model (One independent variable)?**
   For a simple linear regression model, it is enough to only state the coefficients related to the **slope** of the model because the intercept value can be handled by simply adding an extra entry x_0, always equal to 1, to the feature vector x.
   The formula for a simple linear regression with 1 independent variable is:

$$\hat{y} = f_w(x) = w_0 x_0 + w_1 x_1$$

5. **What is cross validation and how does it work?**

   Cross validation is a method used in Data Science to reduce the Overfitting phenomenon when dealing with machine learning in datasets, especially when they are small.
   Cross validation comprises the following steps:

   (a) Split the dataset into k equal parts (folds).
   (b) Train the model on the dataset formed by the k-1 folds (all the parts except for the k*) and validate the results over the k* fold samples remained.
   (c) repeat the process k times, each time choosing a different fold as test-set



(d) Sum the metrics and average them. Compare the results obtained.

Cross Validation is an excellent choice to do hyperparameter tuning of a model over the training data. Another possible choice for the validation of hyperparameters is Leave-One-Out, i.e. a Cross Validation where the size of the folds is 1, althogh could be computationally very expensive.

6. **Mathematically show that the softmax function with two classes (k = 2) is equivalent to the sigmoid function?**

$$\text{Softmax classifier for k classes: } y_k(x) = \frac{exp(w_k^T x)}{\sum_j exp(w_j^T x)}$$

Since we are considering only two classes we have that summation is only over two parameter vectors w1 and w2. If we consider class C1 we may write:

$$y_1(x) = \frac{exp(w_1^T x)}{exp(w_1^T x) + exp(w_2^T x)} = \frac{\frac{exp(w_1^T x)}{exp(w_1^T x)}}{\frac{exp(w_1^T x)+exp(w_2^T x)}{exp(w_1^T x)}} = \frac{1}{1 + exp[(w_1 - w_2)^T x]}$$

7. **As you know, in binomial logistic regression the cost between the true value y and the predicted value ŷ is measured as below:**

$$cost(\hat{y}, y) = \begin{cases} -log(\hat{y}), & \text{if } y = 1 \\ -log(1 - \hat{y}), & \text{if } y = 0 \end{cases}$$

Explain why log is a proper function to compute the cost in logistic regression?
The cost function is equivalent to the MSE, and in the case of the binomial logistic regression the cost function is the following:

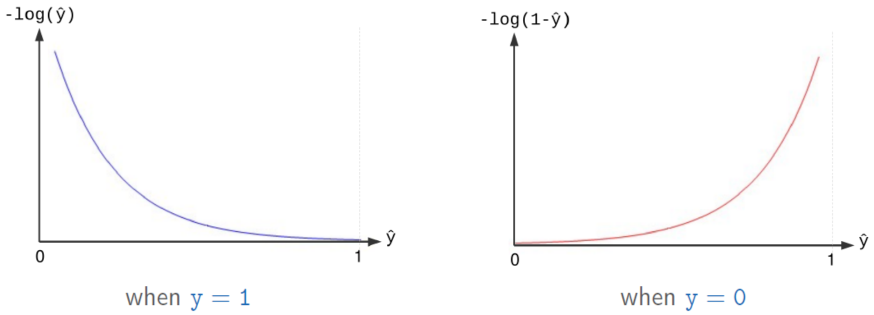


Figure 3: Cost Function Binomial Logistic Regression

$$J(w) = MSE(w) = \frac{1}{m}\sum_{i=1}^{m}(\frac{1}{1 + exp[(-w)^T x^{(i)}]} - y^{(i)})^2$$

This cost function in non convex, and since our aim is to minimize such function the optimization algorithm may converge to a local minimum. To overcome this problem it is better to consider another cost function which should be convex and have the following desired properties:

- Close to 0, if the predicted value ŷ will be close to true value y.
- Large, if the predicted value ŷ will be far from the true value y.

The first cost function instead satisfies all the requirements and it can be used to find the global minimum of a binomial logistic regression model.

8. **How are logistic regression cost, cross-entropy, and negative log-likelihood related?**
The cross-entropy is a measure from the field of information theory which computes the difference among two probability distributions p and q.

$$H(p, q) = -\sum_{j} p_j log(q_j)$$

If we consider p as the true distribution and q as the predicted distribution by a logistic regression model we can obtain a distance measures able to quantify the performance of our model.
In particular if we consider as true and predicted probability distribution the following functions:

$$\text{true probability distribution} = \begin{cases} p(y = 1) = y \\ p(y = 0) = 1 - y \end{cases}$$

$$\text{predicted probability distribution} = \begin{cases} q(y=1) = \hat{y} \\ q(y=0) = 1-\hat{y} \end{cases}$$

We obtain exactly the logistic cost function:

$$H(p,q) = -\sum_j p_j log(q_j) = -(ylog(\hat{y}) + (1-y)log(1-\hat{y})) = cost(y,\hat{y})$$

The likelihood is a statistic function which expresses the probability to observe particular data samples X given as input the model parameters values θ. If samples in X are i.i.d. we can write the following formula:

$$L(\theta) = p(X|\theta) = \sum_{i=1}^{m} p(x^{(i)}|\theta)$$

The negative Log-Likelihood is defined as follow, transforming the likelihood in its negative logarithm to overcome the problem of numerical underflow:

$$-logL(\theta) = -\sum_{i=1}^{m} logp(x^{(i)}|\theta)$$

If we consider the value $\hat{y}^{(i)}$ as the following probability:

$$\begin{cases} p(y^{(i)} = 1|x^{(i)}; w) = \hat{y}^{(i)} \\ p(y^{(i)} = 0|x^{(i)}; w) = 1-\hat{y}^{(i)} \end{cases}$$

The corresponding Negative-Log Likelihood is exactly the same of the logistic regression cost function:

$$-log(L(w)) = -\sum_{i=1}^{m} logp(y^{(i)}|x^{(i)}; w) = -\sum_{i=1}^{m}(y^{(i)}log(\hat{y}^{(i)}) + (1-y^{(i)})log(1-\hat{y}^{(i)}))$$

Therefore, logistic regression cost, cross-entropy, and negative log-likelihood coincide under the assumptions described before.

9. **Explain how a ROC curve works?**
Receiver Operating Characteristic (ROC) curve is a chart which shows the performance of classifier model at all classification threshold, i.e. the threshold set to decide whether elements belong to a class or not.
On the y-axis is plotted the True Positive Rate (TPR) and on the x-axis

the False Positive Rate(FPR). These rates are computed considering the confusion matrix:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

In order to obtain the data points to be plotted in the ROC curve, it is simply required the computation of the TPR and FPR considering the continuous interval of the threshold values $[0, 1]$.
As we can see from the Figure 4, a perfect classifier has a TPR equal to 1 and FPR equal to 0 for every threshold, instead a random classifier will have both rates equal to 50% in all the cases. The more I get closer to the perfect classifier, the better is the Area Under the Curve (AUC) which is the enclosed area by the considered classifier performance and the random classifier.

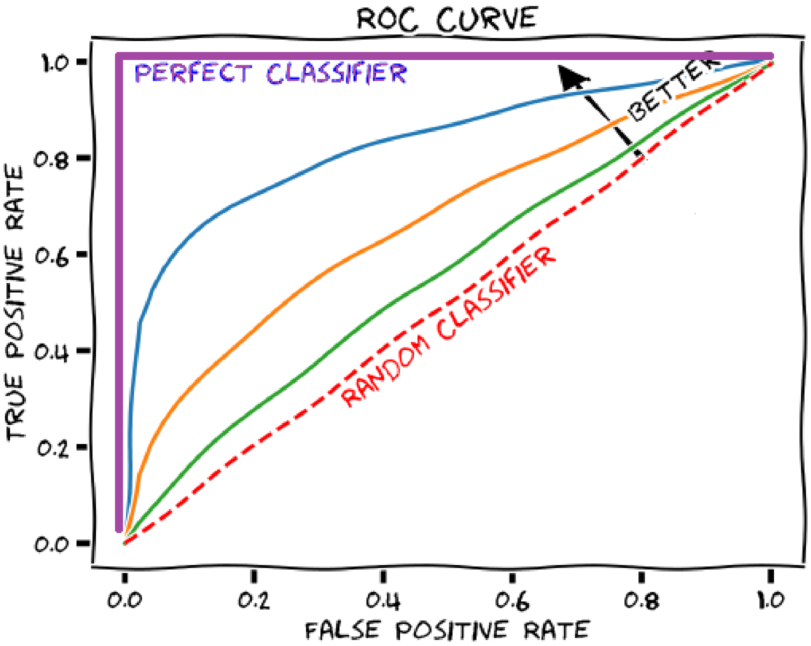


Figure 4: ROC curve, source: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/