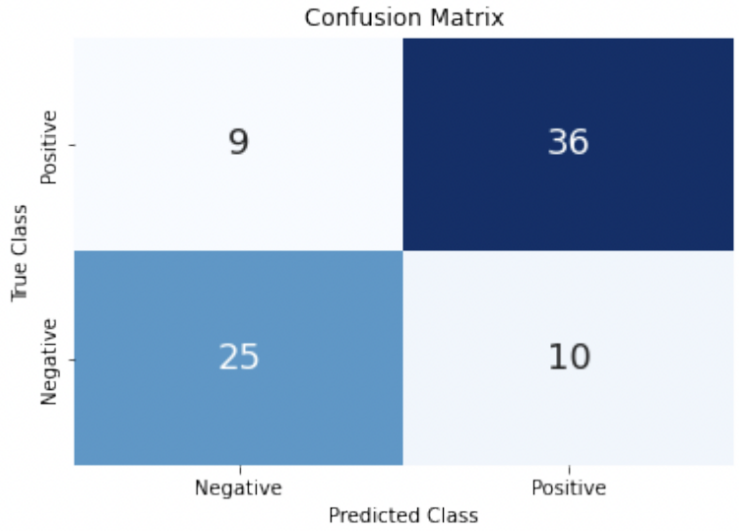
148 HW 4
Margaret Capetz

1. Suppose we have the following confusion matrix outputted from a logistic regression using the probability threshold $P(Y = Positive) \geq t$, i.e. we classify the sample as Positive if $P(Y = Positive)$ is greater than t otherwise we classify as Negative.



(a) Compute the false positive and false negative rates.

(b) How would you expect the confusion matrix to change if we increased t?

a. False positive rate = number of false positives / total number of negatives = 10 / (25+10) = 10/35 = 0.2857 = 28.57%.
False negative rate = number of false negatives / total number of positives = 9 / (9+36) = 0.2 = 20%.

b. If we increased t, there would be less predicted positives since the threshold is greater. Thus, there would be less predicted negatives. The false positive rate would decrease and the false negative rate would increase.

2. *Bayes Theorem.* Consider that you own a small restaurant. You have a smoke detector in your kitchen. The chances that a hazardous fire occurs in the kitchen is pretty rare, say 1%. The smoke alarm is pretty accurate in detecting such fire and it sounds the alarm 99% of the time. However, the alarm is poorly calibrated and it also sounds an alarm sometimes when there is no fire, due to smoke detected from cooking. The accuracy of the smoke alarm under non-fire condition is 90%.

   (a) What is the probability that the smoke detector sounds an alarm?

   (b) Given that you heard the alarm sound, what is the probability that there was actually a fire?

   (c) Comment on how useful the smoke detector is and would you consider replacing it?

2) Baye's Thm

$P(F) = 0.01 \quad P(F') = 0.99$

$P(A|F) = 0.99$

$P(A|F') = 0.1$ (wrong 10% of the time, right 90% of the time)

a) $P(A) = P(F)P(A|F) + P(F')P(A|F')$

$= 0.99 \cdot 0.01 + 0.99 \cdot 0.1$

$= 0.1089 \Rightarrow$ 10.89% chance the alarm goes off.

b) $P(F|A) = \dfrac{P(A|F)P(F)}{P(A)}$

$= \dfrac{0.99 \cdot 0.01}{0.1089}$

$= 0.091 \Rightarrow$ 9.1% chance there was actually a fire.

c) The smoke alarm is not very useful since it is poorly calibrated. It goes off 98.9% of the time, and of those times only 1% of the time it accurately detects a fire. I would consider replacing it, as it has a high false alarm rate.

3. Logistic regression is minimizing the following cross-entropy loss function:

$$L(\beta) = -\sum_{i=1}^{n} y_i \log(\frac{1}{1+e^{-(\beta^T x_i)}}) + (1-y_i)\log(1-\frac{1}{1+e^{-(\beta^T x_i)}})$$

where β is a vector of parameters, n is the number of samples,x_i is a k dimensional data sample, and $y_i \in \{0,1\}$ is a binary variable that represents the class of sample i.

Logistic regression is generally solved using iterative methods. One such method is the gradient descent method where we start with random values for $\{\beta_j^1 : 1 \leq j \leq k\}$ and we update them using the gradient rule

$$\beta_j^{t+1} = \beta_j^t - \eta \frac{dL(\beta)}{d\beta_j^t}$$

for all j such that $1 \leq j \leq k$ where η is the step-size.

Prove that

$$\frac{dL(\beta)}{d\beta_j} = \sum_{i=1}^{n} (\frac{1}{1+e^{-(\beta^T x_i)}} - y_i) x_i^j$$

where x_i^j is the jth element of the ith sample.

$$\frac{\partial L(\beta)}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\left(\sum_{i=1}^{n} y_i \log\left(\frac{1}{1+e^{-(\beta^T x_i)}}\right) + (1-y_i)\log\left(1-\frac{1}{1+e^{-(\beta^T x_i)}}\right)\right)$$

$A = \beta^T x_i$

$B = \frac{1}{1+e^{-A}}$

$$\frac{\partial L(\beta)}{\partial \beta_j} = \frac{\partial L(\beta)}{\partial B} \cdot \frac{\partial B}{\partial A} \cdot \frac{\partial A}{\partial \beta_j}$$

$$\frac{\partial B}{\partial A} = \frac{\partial}{\partial A}\left(\frac{1}{1+e^{-A}}\right) = \frac{e^{-A}}{(1+e^{-A})^2} = \frac{1}{1+e^{-A}} \cdot \frac{e^{-A}}{(1+e^{-A})} = 1 - \frac{1}{1+e^{-A}} = \frac{1+e^{-A}}{1+e^{-A}} - \frac{1}{1+e^{-A}} = B(1-B)$$

$$\frac{\partial A}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}(\beta^T x_i) = x_i^j$$

$$\frac{\partial L(\beta)}{\partial B} = \frac{\partial}{\partial B}\left(-\sum_{i=1}^{n} y_i \log(B) + (1-y_i)\log(1-B)\right)$$

$$= -\sum_{i=1}^{n}\left(\frac{y_i}{B} + \frac{-(1-y_i)}{(1-B)}\right)$$

$$= \sum_{i=1}^{n}\left(\frac{-y_i}{B} + \frac{(1-y_i)}{(1-B)}\right)$$

$$= \sum_{i=1}^{n} \frac{B-y_i}{B(1-B)}$$

$$\frac{\partial L(\beta)}{\partial \beta_j} = \frac{\partial L(\beta)}{\partial B} \cdot \frac{\partial B}{\partial A} \cdot \frac{\partial A}{\partial \beta_j}$$

$$= \sum_{i=1}^{n} \frac{B-y_i}{B(1-B)} \cdot B(1-B) \cdot x_i^j$$

$$= \sum_{i=1}^{n} (B-y_i) x_i^j$$

$$= \sum_{i=1}^{n} \left(\frac{1}{1+e^{-(\beta^T x_i)}} - y_i\right) x_i^j$$

4. In your own words, explain the following types of multi-class classification methods:

   (a) One vs All

   (b) All vs All

   Provide the advantages and disadvantages of each method.

a. Given N classes in a dataset, we generate the N binary classifier models. For example if we have N = 3 classes A, B, and C, we create 3 binary classification models: A vs B,C, B vs A,C, C vs A,B.
   i. Pros: may have an advantage as regards the number of support vectors needed, binary classification can be easily adapted in this way to deal with multi-class classification problems, works well with a small number of classes
   ii. Cons: it doesn't handle binary classification (two classes only) well, the dataset becomes imbalanced because there are many more negative examples than positive ones for each classifier.
b. We generate a binary classifier for each pair of classes, thus for N classes we generate N*(N-1)/2 classes. For the same example of N = 3 classes, we generate 3 * 2 / 2 = 3 binary classification models: A vs B, B vs C, C vs A.
   i. Pros: Handles large datasets with large numbers of classes better than one vs all classification, handles binary (two classes) classification well, creates less models than one vs all, datasets of individual classifiers are balanced when the entire dataset is balanced
   ii. Cons: when N is large, more models are required thus it is more computationally intensive.

5. True or False questions. For each statement, decide whether the statement is True or False and provide justification (full credit for the correct justification).

   (a) For a classification model, positive predictive value is the probability that a model classifies a sample as positive given that the true label of the sample is positive.

   (b) Assume we are working with a multinomial logistic regression such that $P(Y = i|X) = e^{\beta_{0,i}+\beta_{1,i}X}P(Y = K|X)$ for $1 \leq i \leq K-1$. For a dataset with 1 feature and 4 possible class labels, the number of learnable parameters $\beta_{j,i}$ is 8.

   (c) If the log-odds function is modeled as a quadratic, logistic regression can provide a non-linear decision boundary.

   (d) You are building a classifier to detect fraudulent credit card transactions. Your employer states that a 90% success in detection of fraudulent transactions is good enough. You test your model on the next 1000 transactions and get a 97% test accuracy. Therefore, your model is doing much better than what is required.

   (e) For a very good classification model, we expect the confusion table to be dominated by diagonal entries.

a. False, the positive predictive value is the number of true positives over the sum of both true and false positives, which means this measures the probability that the true label is positive given that the model predicts it is positive. This is the converse of the statement.
b. False, there are 4 parameters, not 8. There is one parameter per class.
c. True, as the function determines the shape of the decision boundary.
d. False, the test accuracy should not be the only metric of measurement. In addition to test accuracy, we should measure the precision and recall to verify that the model is outperforming the requirements.
e. True, as we want to maximize the number of True Positives and True Negatives which indicate correct predictions.