16th Baksan School 2019 hackathon additional material: Confusion matrix

G.I. Rubtsov

April 16, 2019

Abstract

The document provides a brief description of the confusion matrix and it's application in the context of classification problem. This is an additional material for Baksan hackathon.

1 Confusion matrix for classification problem

The confusion matrix is defined for a classifier, which we refer here without loss of generality as a neural network (NN). In our case, NN predicts a class of the event, which is one of the four classes (p, He, N, Fe). To estimate the confusion matrix one uses test dataset for which the true class is known. The confusion matrix element Q_{ab} is a probability for NN to predict the class "a" if the true class is "b".

The `sklearn.metrics.confusion_matrix` function calculates the matrix given the predicted and true output values Y^{true} and Y^{pred}.

```
>>> from sklearn.metrics import confusion_matrix
>>> y_true = [2, 0, 2, 2, 0, 1]
>>> y_pred = [0, 0, 2, 2, 0, 2]
>>> confusion_matrix(y_pred, y_true)
array([[2, 0, 1],
       [0, 0, 0],
       [0, 1, 2]])
```

The following normalization of confusion matrix is assumed in this document: the sum of elements of each column is equal to 1:

$$\sum_a Q_{ab} = 1\,. \tag{1}$$

Let us consider the set which is known to have only protons in it. The composition of this set may be represented as a vector $Y^{true} = (1, 0, 0, 0)$. The NN will correctly identify some events and confuse some others, so that the predicted composition will be $Y_a^{pred} = (Q_{10}, Q_{20}, Q_{30}, Q_{40}) = Q_{ab}Y_b^{true}$, where the sum over b is assumed. Let us stop here for the two notices. First, the network will never predict monochromatic composition unless the confusion matrix

off-diagonal elements are all zero. Second, the matrix form of the equation is valid for any true composition:

$$Y_a^{pred} = Q_{ab} Y_b^{true} \,. \tag{2}$$

The Eq. 2 determines the frequency of appearance of each output classes given the true composition. If the matrix Q_{ab} is non-degenerate, it may be inverted.

$$Y_a^{true} = Q_{ab}^{-1} Y_b^{pred} \,. \tag{3}$$

Note, that if NN has no separation power, each of the confusion matrix elements would be equal to 0.25. In this case $\det Q = 0$ and the inversion is not possible. Otherwise, the Eq. 3 may be used for estimating the questioned composition, while the statistical error of the result of Eq. 3 decreases with the growth of the determinant.

2 Search for rare events

Let us now consider a special case of the two-component classifier used for photon search on the dominating proton background. The unknown set contains N_p^{true} proton events and N_γ^{true} photon events. It is known that $N_\gamma \ll N_p$.

The confusion matrix for two-component classification is usually written as:

```
[[TN FN]
[FP TP]]
```

where TN is true negative (number of real protons classified as protons), TP – true positive (real photons classified as photons), FP – false positive (protons classified as photons) and FN – false negative (photons classified as protons).

Given the confusion matrix, one may estimate the number of photon candidates, predicted by NN. These are the sum of false and true positives:

$$N_\gamma^{pred} = \frac{FP}{FP+TN} N_p^{true} + \frac{TP}{TP+FN} N_\gamma^{true} \,. \tag{4}$$

According to Eq. 4, the number of photon candidates, predicted by NN may be significantly larger than the real number of protons in the data set. One may define the false positive rate as $\epsilon_{FP} = \frac{FP}{FP+TN}$ and the recall (or sensitivity) as $\epsilon_\gamma = \frac{TP}{TP+FN}$. The Eq. 4 may be rewritten as:

$$N_\gamma^{pred} = \epsilon_{FP} N_p^{true} + \epsilon_\gamma N_\gamma^{true} \,. \tag{5}$$

The result of the NN for the unknown set is N_γ^{pred}. The rates ϵ_{FP} and ϵ_γ may be calculated with the confusion matrix using the test set. Therefore, one may estimate the true number of photons in the data set with the following equation:

$$N_\gamma^{true} = \frac{N_\gamma^{pred} - \epsilon_{FP} N_p^{true}}{\epsilon_\gamma} \,. \tag{6}$$

The statistical error of the N_γ^{true} is mainly determined by Poisson fluctuation of N_γ^{pred}, which is of the order of $\epsilon_{FP} N_p^{true}$. It may be hence estimated as:

$$\sigma(N_\gamma^{true}) = \frac{\sqrt{\epsilon_{FP} N_p^{true}}}{\epsilon_\gamma} \,. \tag{7}$$

In order to optimize the statistical error given by Eq. 7, one may tune the classifier to decrease false positive rate at the possible cost of some loss of sensitivity.