ML and PR Project Report – Part 5

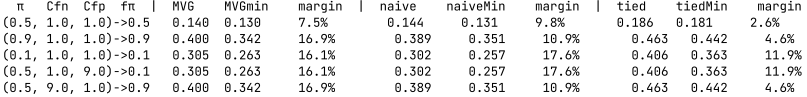
The theme of the current laboratory is the evaluation of different classifiers and different applications, through cost function analysis.

# Evaluating triplets over MVG, Naïve, Tied

The evaluation that we are conducting considers the Detection Cost Functions of our different classifiers trained for optimizing cost. What we are going to do is to compare the recognizers by applying the same prior and cost parameters.

We are going to visualize both the cost related to the actual classifier, whose threshold is dependent on the predicted distribution, and a minimum classifier, that utilizes the same predicted distributions for the samples, but explores an optimal threshold for classification. If the minimum cost is especially lower than the actual cost, then our classifier isn’t really well calibrated, so in order to measure this property of the analysis, I included a margin attribute, that measures the difference in ratio between the actual cost and the minimum cost.

We expect applications that belong to the same equivalence class in terms of normalized DCF (when taking as reference for the normalization the dumbest classifier, which doesn’t regard the training at all and just classifies all the samples with the same class) to yield the same results, and it actually is the case. Applications 2 and 5 have the same effective prior (0.9), and so do 3 and 4 (0.1).



For the 1st application, **MVG (0.140; 0.144)** and **Naïve (0.130; 0.131)** have similar cost, both in terms of actual and minimum cost, while **Tied (0.186; 0.181)** performs worse. In terms of margin though, the Tied assumption seems to be the one that leads to best calibration, by quite a few points **(2.6% for Tied vs 7.5% for MVG and 9.8% for Naïve).**

For the 2nd and 5th applications, again, **MVG (0.400; 0.342)** and **Naïve (0.389; 0.351)** have similar cost, while **Tied (0.463; 0.442)** performs worse. **Tied’s margin (4.6%)** is again the best one, although not as good as before. **Naïve’s margin (10.9%)** didn’t change all that much, while **MVG’s (16.9%)** got much worse.

For the 3rd and 4th applications, once more, **MVG (0.305; 0.263)** and **Naïve (0.302; 0.257)** have similar cost, while **Tied (0.406; 0.363)** performs worse. **Tied’s margin (11.9%)** is again the best one, but it has much worse results than for the other applications. **Naïve’s (17.6%)** too got worse, while **MVG’s (16.1%)** is almost the same as for the =0.9 application, stealing Naïve’s second place.

In general the =0.5 application is always the less costly one, while the =0.9 the most costly, just MVG and Naïve are the less costly ones, with similar results. The margins though, vary in a not really uniform manner for different applications and classifiers, but we can for sure say that the best calibrated classifier is the Tied one, with extremely great results for the =0.5 and =0.9 applications.

# Evaluating effective applications using PCA

We now want to see if PCA is able to improve our results, but that is not the case, as we had already seen when simply analyzing the accuracy of the different classifiers in the previous lab. Just like for the accuracy analysis, m=6 yields identical classification quality as no-PCA, except for Naïve Bayes, meaning that the best result is still no PCA.

There is something curious about the m=5 case though, as the results have almost no difference compared to the m=6 case, except for a much better calibration when considering the complete MVG.

Since we have already observed that applications that have the same effective prior do belong to the same equivalence class in terms of normalized DCF, we can limit our analysis at only three different parameter sets.

# Immagine che contiene testo, ricevuta, Carattere, numero Descrizione generata automaticamente

Immagine che contiene testo, ricevuta, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, ricevuta, Carattere, bianco

Descrizione generata automaticamente

(note that the margin is just a ratio between actual and minimum DCF. It may be better to consider a difference)

Just like for the no-PCA case, =0.5 is always best, while =0.9 is always worst.

MVG tends to be the best classifier, but this time not yielding the same results as Naïve, except for the lower m cases, as Naïve’s predictions, just like we predicted, worsen with the addition of PCA preprocessing.

As for the margin, we don’t have a clear victor anymore.

# Comparing applications with Bayes Error Plots

We now analyze the bayes error plot for the no-PCA application with =0.1, since no-PCA yielded the best results. For reference, the log-odds of the applications from the previous paragraphs are: =0.1 -> -0.95 -- =0.5 -> 0 -- =0.9 -> 0.95.

Immagine che contiene testo, Diagramma, diagramma, linea

Descrizione generata automaticamente

Starting from the center, we confirm that with a of 0.5, we find low marings, i.e. good calibration. In such position, the classifier that falls nearest to its minimum DCF, is the Tied one. There are many more points with good and for some models calibration though, as visible.

We can also confirm that MVG and Naïve perform almost the same (we can almost not distinguish the 2 curves, both for the actual DCF and minimum DCF), and that they barely surpass one another alternately, while Tied is much worse in terms of minimum DCF, but we confirm really good calibration in the origin.

# Conclusions

PCA is detrimental to our dataset. MVG and Naïve have similar performances (except for when applying PCA, at which point Naïve degrades). Tied has good calibration for evenly distributed priors (50/50), but all classifiers, Tied included, present signs of bad calibration as soon as we move away from this prior distribution.