Evaluating and comparing classification algorithms

Pablo González de Prado Salas

Beyond accuracy

Imagine you are presented with a new procedure to predict a health condition during gestation. Relevant problems, such as Down Syndrome, may have an incidence of one case every 1000 births. Suppose that the proposed technique is a perfect predictor of positive cases, meaning that, if the condition is present, it will be predicted with certainty (sensitivity = 100%). Suppose that the technique is also quite robust, in that it only predicts a false positive every 1000 true cases (specificity = 99.9%). (Accuracy = 99.9%, error rate = 0.1%.)

But what can we expect in practice? Should we apply this new technique? Note that our method will (correctly) predict one positive every 1000 children, but also note that it will (incorrectly) predict one false positive every 1000 children. What this means for a expecting mother is that, even if the test is positive, the child is healthy with a 50% chance.

If the condition is severe and tests are not fast enough, this can have enormous emotional implications (even more so if an abortion is considered as a possibility). In some cases (think of cancer) a false positive may actually be harmful, as it may result in needless and aggressive treatments.

Is this an acceptable technique then? It depends on many factors that go beyond our course. The take-away message here is to always think beyond numbers. No single measure is enough to say if an algorithm will be good enough for a proposed task.

Comparing algorithms

As head of a work team, you have developed two alternative algorithms to be used in a classification task. Algorithm A has an accuracy of 75%, and algorithm B has an accuracy of 80%. However, algorithm B is slower than A, so you want to be sure the difference in accuracy is significant.

The evaluation team was tight on budget, and they only managed to provide 100 cases for the evaluation test (where A correctly classified 75 cases and B correctly classified 80 cases). Is the difference significant in this case?

Let us suppose that algorithm B has a "true" accuracy of 75%, but it was lucky during the test. How probable is this case? For convenience, suppose that all algorithms have the same chance of correctly classifying positive and negative examples (which does **not** need to be the case, ask me about this if you have doubts). Let us call p the probability of correct classification, and $q = (1 - p)$ the probability of incorrect classification. If there is only one tuple to be classified, we have one success with p chance and 1 fail with q chance,

$$p \qquad q$$

for two tuples we have two successes with p^2 chance, one success with $2p \times q$ chance and two fails with q^2. Note there are two different ways of having one fail and one success!

$$p^2 \qquad 2p \times q \qquad q^2$$

We can keep counting chances for different total number of tuples:

$$\begin{array}{cccc}
p & q & & \\
p^2 & 2p \times q & q^2 & \\
p^3 & 3p^2 \times q & 3p \times q^2 & q^3 \\
\cdots & \cdots & \cdots & \\
\left(\begin{array}{c} n \\ 0 \end{array} \right) p^n \times q^0 & \left(\begin{array}{c} n \\ 1 \end{array} \right) p^{n-1} \times q^1 & \cdots & \left(\begin{array}{c} n \\ n \end{array} \right) p^0 \times q^n
\end{array}$$

That is to say, the chance of correctly classifying m out of n tuples (if the constant probability of correct classification is *p*) is

$$\left(\begin{array}{c} n \\ m \end{array} \right) p^{n-m} \times q^m.$$

That should be enough to answer our question: how likely is it that with true accuracy 75% algorithm B scored 80/100 correct classifica-

tions? The quick, misleading answer is "~5%". But we are actually interested in the chance that the accuracy might deviate 5 units from the expected 75/100 (that is, how likely it would be to score 70 or 71 or 72 or ... or 80). Clearly we could just add all probabilities, but things are going to get out of hand quickly if we follow that path, specially if we want to consider more than just 100 test tuples. (The main problem is that we are going to end up having divisions of huge numbers over huge numbers.)

To avoid this problem and to get a faster estimation (instead of manually adding all the probabilities), we may approximate our binomial probabilities using a normal distribution. For n test examples, the binomial distribution has an expected average of $\mu = n \times p$, and you can trust that the variance is $\sigma^2 = npq$ (or look into it if you don't trust me). With these two parameters we can define our corresponding normal distribution:

$$\mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

However, a normal distribution is a continuous function. How do we find then the chance of, say, 5 successes? A good approximation is to integrate the normal distribution between 4.5 and 5.5:

$$P\left(n\right) \approx \int_{n-0.5}^{n+0.5} \mathcal{N}\left(\mu, \sigma^2\right) dx.$$

Now we have a fast way of predicting the chance of finding a result within a desired range. First we will define the normal distribution by finding the average and variance, and then we will integrate between the limits we need (for example, we would integrate between 69.5 and 80.5 to find the cumulative probability of correctly classifying between 70 and 70 tuples).

You may argue that having to integrate the normal distribution is not the fastest and most convenient solution. Fortunately the Internet comes to save the day. For example, in this address you only need to provide the average, variance and limits to get your result. There are many other options, such as Wolfram Alpha or coding a fast example in your language of choice.

If you have come all this way, I can finally give you some answers. If we have 100 test tuples and 75% accuracy, the chance that the number of correct classified tuples is **not** within 70 and 80 is:

$$P_{100}\left(\textbf{not } 70-80\right) \approx 20.4\%$$

which is not a small probability! This means there is a 20% chance that the measured accuracy, with 100 test tuples, deviates more than $\pm 6.7\%$ (~6.7% is 5 units with respect to the expected 75). Can you say in this case that an algorithm with measured accuracy of 75% is truly better than another with measured accuracy of 77%? (The answer is: no, but in this case it should be easy to get a better, more reliable evaluation.)

Here you have other calculations, so you can see the dramatic effect of increasing the number of tuples in the test data. These probabilities explore the possibility of having results outside a bracket of ± 5 and ± 1 percentage points around our expected accuracy of 75%:

$$P_{100}\left(\textbf{not } 70-80\right) \approx 20.4\%$$
$$P_{100}\left(\textbf{not } 74-76\right) \approx 72.9\%$$

$$P_{1000}\left(\textbf{not } 700-800\right) \approx 0.0002\%$$
$$P_{1000}\left(\textbf{not } 740-760\right) \approx 44.32\%$$

$$P_{10000}\left(\textbf{not } 7000-8000\right) \approx 0\%$$
$$P_{10000}\left(\textbf{not } 7400-7600\right) \approx 0.0203\%$$