Calibration of Machine Learning Models.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
(+34) 963877007 (ext. 73585, 73586, 83505, 73537), (+34) 963877359
{jorallo, mramirez, cferri, abella}@dsic.upv.es

Calibration of Machine Learning Models

ABSTRACT

Evaluation of machine learning methods is a crucial step before application, because it is essential to assess how good a model will behave for every single case. In many real applications, not only the "total" or the "average" of the error of the model is important but it is also important to know how this error is distributed or how well confidence or probability estimations are made. However, many machine learning techniques are good in overall results but have a bad distribution /assessment of the error.

In these cases, calibration techniques have been developed as postprocessing techniques which aim at improving the probability estimation or the error distribution of an existing model.

In this chapter, we present the most usual calibration techniques and calibration measures. We cover both classification and regression, and we establish a taxonomy of calibration techniques, while then paying special attention to probabilistic classifier calibration.

INTRODUCTION

One of the main aims of machine learning methods is to build a model or hypothesis from a set of data (also called evidence). After this learning process, it is usually necessary to evaluate the quality of the hypothesis as precisely as possible. For instance, if prediction errors have negative consequences in a certain application domain of a model (for instance, detection of carcinogenic cells) it is important to know exactly the accuracy the model has. Therefore, the model evaluation stage is crucial for the real application of the machine learning techniques. Generally, the quality of predictive models is evaluated using a training set and a test set (which are usually obtained by partitioning the evidence into two disjoint sets), or using some kind of cross-validation or bootstrap if more reliable estimations are desired. These evaluation methods work for any kind of estimation measure. It is important to mention that different measures can be used depending on the model. The most common measures are, for classification models, accuracy (or, inversely error), f-measure, or macro-average. In probabilistic classification, besides the percentage of correctly classified instances, other measures like logloss, mean squared error (MSE) or Brier's score, or area under the ROC curve (AUC) are used. For regression, the most common ones are to use MSE or the mean absolute error (MAE).

With a same result for a quality metric (e.g. MAE), two different models might have a different error distribution. For instance, a regression model R_1 always predicting the true value plus 1 has a MAE of 1 but it is different to a model R_2 that predicts the true value for each n examples but one, where the error for this one is n. The first seems to be more reliable or stable, or in other words, its error is more predictable. Similarly, with the same result for a quality metric (e.g. accuracy), two different models might have different error assessment. For instance, a classification model C_1 which is correct 90% of the cases with confidence 0.91 for every prediction is preferable to model C_2 which is correct 90% of the cases with confidence 0.99 for every prediction. The error self-assessment, i.e. the purported confidence is more accurate in the first case than in the second one.

In both cases above, an overall picture of these results, i.e. an empirical behaviour of how it behaves, is helpful to improve its reliability or confidence in many cases. In the first case, the regression model R_1 which always predicts the true value plus 1 is clearly uncalibrated, since predictions are usually 1 unit above the real value. Subtracting 1 unit to all prediction would calibrate R_1. Interestingly, R_2 is calibrated in the same way. For the second case, a global calibration

requires confidence estimation to be around 0.9 since the models are right 90% of the time.

So, calibration might be understood in many ways, but it is usually built around two issues: how error is distributed, and how self-assessment (confidence or probability estimation) is performed. Although both ideas can be applied for both regression and classification, the first one has been mainly discussed for regression and the second one for classification.

Both are closely related, since for a regular, predictable model such as R_1 the one which always predicts the true value plus 1, it is much easier to estimate a probability (since it is a continuous value, a probability density function). In this case, the probability estimation can be just a density function which includes all the probability to the interval between the predicted value minus 1 and the predicted value.

Estimating probabilities or confidence values is crucial in many real applications. For example, if we have accurate probabilities, decisions can be made with a good assessment of risks and costs, using utility models or other techniques from decision making. Additionally, their integration with other models (e.g. multiclassifiers) or with previous knowledge becomes more robust. In classification, probabilities can be understood as confidence degrees, especially in binary classification, thus accompanying every prediction with a reliability score (DeGroot & Fienberg, 1982). Regression models might accompany predictions by confidence intervals or by probability density functions.

In this context, and instead of redesigning any existing method to directly obtain good probabilities or a better error distribution, some calibration techniques have been developed to date. A calibration technique is any postprocessing technique which aims at improving the probability estimation or to improve error distribution of a given predictive model. Given a general calibration technique, we can use it to improve any existing machine learning method: decision trees, neural networks, kernel methods, instance-based methods, Bayesian methods, etc., but it can also be applied to hand-made models, expert systems or combined models

Depending on the task, different calibration techniques can be applied, and the definition of calibration can be stated more precisely. The most usual calibration techniques are listed below, including different names we give them in order to clarify the rest of this chapter, as well as a type code:

- TYPE CD. Calibration techniques for discrete classification ("(class) distribution calibration in classification" or simply "class calibration"): a typical decalibration arises when the model predicts examples of one or more classes in a proportion which does not fit the original proportion, i.e., the original class distribution. In the binary case (two classes) it can be expressed as a mismatch between the expected value of the proportion of classes and the actual one. For instance, if a problem has a proportion of 95% of class 'a' and 5% of class 'b', a model predicting 99% of class 'a' and 1% of class 'b' is uncalibrated, although it could have a low error (ranging from 4% to 5%) . This error distribution can be clearly seen on a confusion or contingency table. So, class calibration is defined as the degree of approximation of the true or empirical class distribution with the estimated class distribution. The standard way to calibrate a model in this way is by changing the threshold that determines when the model predicts ‘a’ or ‘b’, making this threshold stricter with class ‘a’ and milder with class ‘b’ to balance the proportion. Note that this type of calibration, in principle, might produce more error. In fact, it is usually the case when one wants to obtain a useful model for problems with very imbalanced class distribution, i.e. the minority class has very few examples. Note that we are usually interested in a match between global proportions, but this calibration can also be studied and applied locally. This is related to the problem of "repairing concavities" (Flach & Wu, 2005).

- TYPE CP. Calibration techniques for probabilistic classification ("probabilistic calibration for classification"): a probabilistic classifier is a classifier which accompanies each prediction with a probability estimation. If we predict that we are 99% sure, we should expect to be right 99% of the times. If we are only right 50% of the times, this is not calibrated because our estimation was too optimistic. Similarly, if we predict that we are only 60% sure, we should expect to be right 60% of the times. If we are right 80% of the times, this is not calibrated because our estimation was too pessimistic. In both cases, the expected value of the number or proportion of right guesses (in this case the probability or the confidence assessment) does not match the actual value. Calibration is then defined as the degree of approximation of the predicted probabilities to the actual probabilities. More precisely, a classifier is perfectly calibrated if for a sample of examples with predicted probability p, the expected proportion of positives is close to p. Note that accuracy and calibration, although dependent, are very different things. For instance, a random classifier (a coin tosser) which always assigns 0.5 probability to their predictions is perfectly calibrated. On the other side, a very good classifier can be uncalibrated if correct positive (resp. negative) predictions are accompanied by relative low (resp. high) probabilities. Also note that good calibration usually implies (except from the random coin tosser) that estimated probabilities are different for each example. For some examples, confidence will be high and for other more difficult ones, confidence will be low. This implies that measures to evaluate this type of calibration must evaluate agreement between the expected value and the real value in a local way, by using partitions or bins of the data.

- TYPE RD. Calibration techniques to fix error distribution for regression ('distribution calibration in regression'): in this case the errors are not distributed regularly along the output value. The error is concentrated in the big values or it is gone over to positive or negative values. The expected value which should be close to the actual value can be defined in several ways. For instance, the expected value of the estimated value (y_{est}) should be equal (or close) to the real value (y), i.e. $E(y_{est}) = E(y)$ or, equivalently, $E(y_{est} - y) = 0$. In the example R_1 above, $E(y_{est}) = E(y) + 1$. The mean error (its expected value) would be 1 and not 0. Another equation that shows that a model might be uncalibrated is the expected value of the quotient between the estimated value and the real value, $E(y_{est} / y)$ which should be equal or close to 1. If this quotient is greater than one, the error used to be positive for high values and negative for low values. Typically, these problems are detected and penalised by typical measures for evaluating regression models, and many technique (e.g. linear regression), generate calibrated models (at least in these two aspects mentioned above). Other kind of more sophisticated techniques or, more frequently, hand-made models, might be uncalibrated and might require a calibration.

- TYPE RP. Calibration techniques to improve probability estimation for regression ('probabilistic calibration for regression'): This is a relatively new area (Carney & Cunningham, 2006) and is applicable when continuous predictions are accompanied or substituted by a probability density function (or, more restrictively, confidence intervals). This kind of regression models are usually referred as "density forecasting" models. Instead of saying that temperature is going to be 23.2º Celsius, we give a probability density function from which we can calculate that the probability of the temperature to be between 21º and 25º is 0.9 and the probability of the temperature to be between 15º and 31º is 0.99. If our predictions are very accurate, density functions (and hence confidence intervals) should be narrower. If our predictions are bad, density functions should be broader, in order to approximate the estimated probabilities to the real probabilities. As in the type CP, a good calibration requires in general that these density functions are particular for each prediction, i.e., for some cases where the confidence is high, confidence intervals will be narrower. For difficult cases, confidence intervals will be broader. As in the type CP, measures to evaluate this type of calibration must evaluate agreement between the expected value and the real value in a local way, by using partitions or bins of the data.

Table 1 summarises these four types of calibration.

TYPE	Task	Problem	Global/Local	What is calibrated?
CD	Classification	Expected class distribution is different from real class distribution	Global or local	Predictions
CP	Classification	Expected/estimated probability of right guesses different from real proportion.	Local	Probability/confidence
RD	Regression	Expected output is different from real average output.	Global or local	Predictions
RP	Regression	Expected/estimated error confidence intervals or probability density functions are too narrow or too broad.	Local	Probability/confidence

Table 1. A taxonomy of calibration problems.

Note that types CD and RD necessarily must modify predictions in order to calibrate the results. In type CD, if we move the class threshold, some predictions change. In RD if we try, let us say, to reduce high values and increase low values, predictions also change. In contrast, for types CP and RP, calibration can be made without (necessarily) modifying predictions: only confidence assessments or probabilities need to be touched. For CP, in particular, these kinds of calibrations are known as *isotonic*. Consequently, some measures as average error will not be affected by these two types of calibrations.

Additionally, since we want to improve calibration, we need measures to evaluate this improvement. A calibration measure is any measure which is able to quantify the degree of calibration of a predictive model. For each type of calibration model, some specific measures are useful to evaluate the degree of calibration, while others are only partially sensitive or completely useless. For instance, for CP, the most common measure, accuracy (or % of errors), is completely useless. For RP, the most common measure, MSE, is completely useless. We will review some of these calibration measures in the following section.

For all the types in Table 1, type CP is the one which has devoted more attention recently. In fact, for many researchers in machine learning, the term "calibration" usually refers to this type, without the need of specifying that there are other types of calibration. Additionally, this is the type which has developed more techniques and more specific measures. Furthermore, regression techniques and measures have been traditionally developed to obtain calibrated models, so less improvement is expected from calibration techniques. For this reason, we will devote much more space to classification, and very especially to type CP.

Overall, in this chapter we give a general overview about calibration and review some of the most-known calibration evaluation measures and calibration methods which have been proposed for classification and regression. We conclude analysing some open questions and challenges which can constitute the research on calibration in the future.

CALIBRATION EVALUATION MEASURES

As mentioned in the introduction, a calibration measure is any measure which is able to quantify the degree of calibration of a classifier. As we can see in Table 2, many classical quality metrics are not useful to evaluate calibration techniques. In fact, new and specific measures have been derived or adapted to evaluate calibration, especially for types CP and RP.

TYPE	Calibration measures	Partially sensitive measures	Insensitive measures
CD	Macro-averaged accuracy, proportion of classes.	Accuracy, mean F-measure, ...	Area Under the ROC Curve (AUC), MSE^p, Logloss, ...
CP	MSE^p, LogLoss, CalBin, CalLoss		AUC, Accuracy, mean F-measure, ...
RD	Average error, Relative error	MSE^r, MAE, ...	
RP	Anderson-Darling (A2) test		Average error, relative error, MSE^r, MAE, ...

Table 2. Calibration measures (second column) for each type of calibration problem. On the third and fourth columns we show measures which are partially sensitive (but not very useful in general) or completely insensitive to each type of calibration.

The second column in the above table shows the calibration measures. This does not mean, though, that these measures *only* measure calibration. For instance, for type CP, CalBin and CalLoss only evaluate calibration, while MSE or Logloss evaluate calibration and other components at the same time. We will refer to these two types of measures as pure and impure calibration measures, respectively. Pure calibration measures have the risk that a classifier which always predicts the positive prior probability is perfectly calibrated according to these measures. Following with the type CP, some other metrics are insensitive to calibration, such as qualitative measures (accuracy, mean F-measure, etc.), provided the calibration function is also applied to the threshold, or measures of ranking (such as AUC), provided that calibration modifies the value of the probabilities but not their order. This is the reason-why calibration has emerged as an important issue, since for many traditional quality metrics, calibration issues are completely disregarded. Hence, many machine learning techniques generate uncalibrated models.

Note that we use two different terms for Mean Square Error, MSE^p and MSE^r. The reason-why is that the first one is used for classification and compares the estimated probabilities with the actual probability (0 or 1), while the second one is used for regression and compares two continuous values.

From the measures in the second column, Macro-averaged accuracy, proportion of classes, MSEp, LogLoss, CalBin, CalLoss, Average error, Relative error, Anderson-Darling (A2) test, some of them are very well-known and do not need any further definition, but a few words: macro-averaged accuracy is the average of the partial accuracies for each class, the proportion of classes is computed for the predictions on a dataset and can be compared with the real proportion. Average error and relative error are well-known in regression. Consequently, we will devote the rest of this section to explain MSE^p, LogLoss, CalBin, CalLoss for type CP and Anderson-Darling (A2) test for RP.

We will first start with the measures that are applicable to probabilistic classifiers. We use the following notation. Given a dataset T, n denotes the number of examples, and C the number of classes. $f(i, j)$ represents the actual probability of example i to be of class j. We assume that $f(i, j)$

always takes values in {0,1} and is strictly not a probability but an indicator function. With $n_j = \sum_{i=1}^{n} f(i,j)$, we denote the number of examples of class *j*. *p*(*j*) denotes the prior probability of class *j*, i.e., $p(j) = n_j / n$. Given a classifier, $p(i,j)$ represents the estimated probability of example *i* to be of class *j* taking values in [0,1].

With these definitions, we can define the measures as follows.

Mean Squared Error

Mean Squared Error (MSE) is a measure of the deviation from the true probability. We have used MSE^p to distinguish this measure with the homonym used in regression. In classification it is also know as Brier Score. It is defined as

$$MSE = \sum_{j=1}^{C} \sum_{i=1}^{n} \frac{(f(i,j) - p(i,j))^2}{n \times C}$$

Although originally MSE is not a calibration measure, it was decomposed in (Murphy, 1972) in terms of calibration loss and refinement loss. An important idea of the decomposition is that data is organised into bins, and the observed probability in each bin is compared to the predicted probability or to the global probability. As we will see, some kind of binning will be present in many calibration methods and measures. The calibration component is the only one which is affected by isotonic calibrations, since discrimination and uncertainty components are not modified if probabilities are calibrated in such a way that the order is not modified (i.e. isotonic), since bins will not be altered. For the decomposition, *T* is segmented in *k* bins.

$$MSE = \frac{\sum_{j=1}^{C} \sum_{l=1}^{k} \sum_{i=1, i \in l}^{n_l} n_l \times \left(p(i,j) - \overline{f}(i,j)\right)^2 - \sum_{l=1}^{k} n_l \times \left(\overline{f}_l(i,j) - \overline{f}(j)\right) + \overline{f}(j) \times \left(1 - \overline{f}(j)\right)}{n \times C}$$

where $\overline{f}_l(i,j) = \sum_{i=1, i \in l}^{n_l} \frac{f(i,j)}{n_l}$ and $\overline{f}(j) = \sum_{i=1}^{n} \frac{f(i,j)}{n}$. The first term measures the calibration of the classifier while the rest of the expression measures other components usually grouped under the term "refinement ".

LogLoss

Logloss is a similar measure, and it is also known as probabilistic cost or entropy. It is very related with the Kullback-Leibler distance between the real model and the inferred model (Good, 1952; Good, 1968; Dowe, Farr, Hurst, & Lentin, 1996). It is defined as follows:

$$Logloss = \sum_{j=1}^{C} \sum_{i=1}^{n} \frac{(f(i,j) \times \log p(i,j))}{n}$$

Calibration by Overlapping Bins

One typical way of measuring classifier calibration is that the test set must be split into several segments or bins, as the MSE decomposition shows (although MSE does not need to use bins to be computed). The problem of using bins is that if too few bins are defined, the real probabilities are not properly detailed to give an accurate evaluation. If too many bins are defined, the real probabilities are not properly estimated. A partial solution to this problem is to make the bins overlap.

A calibration measure based on overlapping binning is CalBin (Caruana & Niculescu-Mizil, 2004). This is defined as follows. For each class, we must order all cases by predicted $p(i, j)$, giving new indices i^*. Take the 100 first elements (i^* from 1 to 100) as the first bin. Calculate the percentage of cases of class j in this bin as the actual probability, $\hat{f}_j$. The error for this bin is $\sum_{i^* \in 1..100} \left| p(i,j) - \hat{f}_j \right|$. Take the second bin with elements (2 to 101) and compute the error in the same way. At the end, average the errors. The problem of using 100 as Caruana and Niculescu-Mizil (2004) suggest is that it might be a much too large bin for small datasets. Instead of 100 we might set a different bin length, $s = n/10$, to make it more size-independent. Formally:

$$CalBin(j) = \frac{1}{n-s} \sum_{b=1}^{n-s} \sum_{i^*=b}^{b+s-1} \left| p(i^*, j) - \frac{\sum_{i^*=b}^{b+s-1} f(i^*, j)}{s} \right|$$

Calibration Loss

In (Fawcett & Niculescu-Mizil, 2007) and, independently, in (Flach & Matsubara, 2007), the relationship between the AUC-based measures,and ROC analysis in general, with calibration was clarified. The receiver operating characteristic curve (ROC curve) is a graphical depiction of classifiers based on their performance, It is generally applied to binary classifiers. The ROC space is a two-dimensional graph in which the True positive rate (the fraction of positives correctly classified or *tp rate*) is plotted on the *Y* axis and the False positive rate (the fraction of negatives incorrectly classified or *fp rate*) is plotted on the *X* axis.. Each discrete classifier produces an (*fp rate, tp rate*) pair that corresponds to a single point in the ROC space. Probabilistic classifiers provide a value (probability or score) that represents the degree to which an instance belongs to a class. In combination with a threshold, the classifier can behave as a binary classifier assigning a class (for instance, positive) if the produced score is above the threshold and the other class (negative) otherwise. In these cases, each threshold produces one point in the ROC space, and drawing a line crossing all the points, a ROC curve is generated. The area under a ROC curve is abbreviated as AUC. "The AUC has an important statistical property: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. This is
equivalent to the Wilcoxon test of ranks" (Fawcett, 2006, p. 868). So, in some way, the AUC is a class separability or instance ranking measure because it evaluates how well a classifier ranks its predictions.

A perfectly calibrated classifier always gives a convex ROC curve. However, a classifier can produce very good rankings (high AUC), but probabilities might differ from the actual probabilities. A method for calibrating a classifier is to compute the convex hull or, equivalently, to use isotonic regression. Flach and Matsubara (2007) derive a decomposition of the Brier Score into calibration loss and refinement loss. Calibration loss is defined as the mean squared deviation from empirical probabilities derived from slope of ROC segments.

$$CalLoss(j) = \sum_{b=1}^{r_j} \sum_{i \in s_{j,b}} \left(p(i,j) - \sum_{i \in s_{j,b}} \frac{f(i,j)}{\left| s_{j,b} \right|} \right)$$

where *rj* is the number of segments in the ROC curve for class j, i.e. the number of different estimated probabilities for class j: |{p(i, j)}|. Each ROC segment is denoted by $s_{j,b}$, with $b \in 1 \ldots r_j$, and formally defined as:

$$s_{j,b} = \left\{ \in 1 \ldots n \middle| \forall k \in 1 \ldots n : p(i,j) \geq p(k,j) \wedge i \notin s_{j,d}, \forall d < b \right\}$$

Anderson-Darling (A^2) test

From type CP, we move now to type RP, where the task is usually referred to as density forecasting, where instead of predicting a continuous value, the prediction is a probability density function. Evaluating this probability density function in terms of calibration cannot be done with typical measures such as MSEr, relative quadratic error or other classical measures in regression. (Carney & Cunningham 2006) adapt an old measure, the Anderson-Darling (A^2) test over the probability integral transform, as a measure of pure calibration. This measure is used to evaluate whether the probability density functions estimated by the regression model are accurate. For the specific definition, we refer the reader to (Carney & Cunningham, 2006).

CALIBRATION METHODS FOR TYPE CD

In the case of discrete classification, the best way to know whether a model is uncalibrated according to the number of instance per class (type CD) is to analyse the confusion matrix. The confusion matrix is a visual way of showing the recount of cases of the predicted classes and their actual values. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another). For example, if there are 100 test examples and a classifier, an example of a confusion matrix with three classes *a*, *b* and *c* could be as follows:

		Real		
Predicted		*a*	*B*	*c*
	a	20	2	3
	b	0	30	3
	c	0	2	40

In this matrix from 100 examples, 20 were from class *a* and all of them were well classified, 34 were from class *b*, 30 of them were well classified as *b*, 2 were misclassified as *a* and 2 were misclassified as *c*. Finally, 46 of the examples were from class *c*, 40 of them were well classified as *c*, 3 were misclassified as *a* and 3 were misclassified as *b*. If we group by class, we have a proportion of 20, 34, 46 for the real data, and a proportion of 25, 33, 42 for the predicted data. As we can see, these proportions are quite similar and, therefore, the classifier is calibrated regarding the original class distribution. On the contrary, the following matrix can be considered:

		Real	
Predicted		*a*	*B*
	a	60	2
	b	40	23

In this matrix the proportion of real data are 100, 25, while the proportion of predicted data are 62, 63. So, in this case the model is uncalibrated. One question would be if this type of disproportion is quite common. The answer is that this situation is very common, and normally the disproportion used to be in favour of the majority classes. The second question is whether there are any techniques to solve the problem after obtaining the model. And the answer is 'yes', in general. To do so, the predictions of the models must be accompanied by probabilities or reliability values (or simply, scores). In this case the threshold that splits into the classes can be changed.

We can consider the technique presented by (Lachiche & Flach, 2003). That work is specialized in Bayesian classifiers, but the technique can be applied to any classification model which accompanies its predictions by probabilities or reliabilities. A naïve Bayes classifier estimates the probabilities for each class independently. So, for example, we can have the following probabilities for each class: $p(a|x) = 0{,}2$ and $p(b|x) = 0{,}05$, and there are no more classes, so, the sum of the probabilities is not 1. In general, the naïve Bayes classifiers assigns very low probabilities, because the probability is the product of several factors that, at the end, reduce the absolute values of the probabilities. Nevertheless, this is not the problem. The problem is that the decision rule that is used to apply the model is the next one:

If $p(a|x) > p(b|x)$ then predicts a
else predicts b

The previous rule has as consequence that the result is not calibrated in most of the cases. It is possible that the previous rule produces much more examples of the class a (or vice versa) that there were in the original distribution. The solution to this problem is to estimate a threshold fitted to the original distribution.

If there are only two classes the solution is very easy, we can calculate a ratio of the two probabilities: $r = p(a|x)/p(b|x)$. This ratio comes from 0 to infinite. We can normalise it between 0 and 1 with a sigmoid, if we want. The aim is to obtain a threshold u with the test set where the distribution of the model will be similar to the original distribution. So, the rule changes to:

If $r > u$ then predicts a
else predicts b

Lachiche and Flach (2003) shows that only with an adjustment like this (in that work the threshold adjustment is based in the ROC analysis and it is extended to multiclass) the results of the models can be improved significantly. In particular, from 25 analyzed datasets, this simple optimization improved significantly the accuracy in 10 cases and was reduced only in 4 of them.

Apart from that simple approximation, there are other works where the optimum threshold fitted to the original distribution is calculated.

CALIBRATION METHODS FOR TYPE CP

Another case is the calibration of probabilistic classification models (type CP) and it requires more sophisticated techniques. In this case, the aim is that when the model predicts that the probability of the class a is 0.99, this means that the model is more confident that the class is a that when it is predicted 0.6. Determining the reliability of a prediction is fundamental in a lot of applications: diagnosis, instance selection and model combination.

Apart from the measures introduced in previous sections (MSE, logloss, CalBin and CalLoss), the fundamental tool to analyze the calibration of this type of models is the reliability diagrams (DeGroot & Fienberg, 1982). In those diagrams, the prediction space is discretised into 10 intervals (from 0 to 0.1, from 0.1 to 0.2, etc.). The examples whose probability is between 0 and 0.1 go into the first interval, the examples between 0.1 and 0.2 go into the second, etc. For each interval, the mean predicted value (in other words, the mean predicted probability) is plotted (x axis) in front of the fraction of positive real cases (y axis). If the model is calibrated the points will be near to the diagonal.

The following Figure 1 shows an example of an uncalibrated model and one calibrated model.

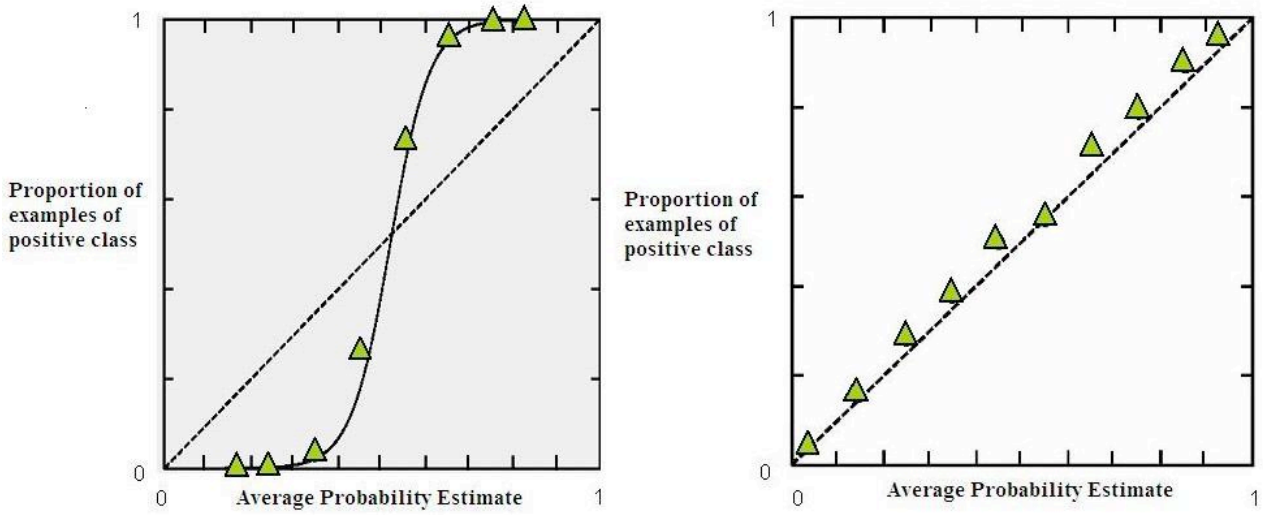

Figure 1. Reliability diagrams. Left: uncalibrated model. Right: calibrated model.

For example, in the left model there are not cases with predicted probability lower than 0.1. The next interval, where the examples have an assigned probability between 0.1 and 0.2 for the positive class (with a mean of 0.17) there are not examples of the positive class. So, these predictions have too high estimated probability. It should be nearer to 0, instead of nearer to 0.17. On the contrary, if we goes to the end of the curve, we will see that the examples with assigned probabilities between 0.7 and 0.8, all of them are from the positive class. Probably, they should have higher probability, because they are surer cases.

On the other hand, in the model on the right, we can see that the correspondence is righter: there are probabilities distributed from 0 to 1 and, moreover, they used to be the same as the percentage of examples.

There are several techniques that can calibrate a model like the left one and transform it in a model like the right one. The most common are: binning averaging, isotonic regression and Platt's method. The objective of these methods (as a postprocessing) is to transform the original estimated probabilities (scores can also be used (Zadrozny & Elkan, 2002)) $p(i, j)$ into calibrated probability estimates $p^*(i, j)$. It is important to remark that all of these general calibration methods can only be used (directly, without approaches) in binary problems, because all of them use the sorted estimated probability to calculate the calibrated probability.

When the calibration function is monotonically nondecreasing (also called isotonic). Most calibration methods presented in the literature are isotonic. This makes it reasonable to use MSE or LogLoss as measures to evaluate calibration methods, since the 'separability components' are not affected. This is clearly seen through the so-called "decompositions of the Brier score" (Sanders, 1963; Murphy, 1972) included in a previous section in this chapter.

Binning Averaging

The first calibration method is called binning averaging (Zadrozny & Elkan, 2001) and consists in sorting the examples in decreasing order by their estimated probabilities and dividing the set into k bins (i.e. subsets of equal size). Then, for each bin l, $1 \leq l \leq k$, the corrected probability estimate for a case i belonging to class j, $p^*(i,j)$, is the proportion of instances in l of class j. The number of bins must be small in order to reduce the variance of the estimates. In their paper, Zadrozny and Elhan fixed k=10 in the experimental evaluation of the method.

Example: Consider the following training set sorted by its probability of membership to positive class grouped in 5 bins.

bin	instance	score
1	e_1	0.95
	e_2	0.94
	e_3	0.91
	e_4	0.90
2	e_5	0.87
	e_6	0.85
	e_7	0.80
	e_8	0.76
3	e_9	0.70
	e_{10}	0.66
	e_{11}	0.62
	e_{12}	0.62
4	e_{13}	0.51
	e_{14}	0.49
	e_{15}	0.48
	e_{16}	0.48
5	e_{17}	0.45
	e_{18}	0.44
	e_{19}	0.44
	e_{20}	0.42

Then, if a new example is assigned a score of 0.68, then it belongs to bin 3 and its corrected probability is $\frac{0.70 + 0.66 + 0.62 + 0.62}{4} = 0.65$.

This is just how binning averaging works.

Isotonic Regression (PAV)

A slightly more sophisticated technique also for two-class problems is isotonic regression. (Ayer, Brunk, Ewing, Reid, & Silverman, 1955) presented a pair-adjacent violators algorithm (PAV) for calculating the isotonic regression. The idea is that calibrated probability estimates must be a monotone decreasing sequence, i.e., $p_1 \geq p_2 \geq ... \geq p_n$. If it is not the case, the PAV algorithm each time that a pair of consecutive probabilities, $p(i,j)$ and $p(i+1,j)$, does not satisfy the above property $p(i,j) < p(i+1,j)$ replaces both of them by their probability average, that is:

$$p^*(i,j) = p^*(i+1,j) = \frac{p(i,j) + p(i+1,j)}{2}$$

This process is repeated (using the new values) until an isotonic set is reached.
Example. The next table shows in the first column the initial scores of one dataset composed by 10 examples. The following columns represent the results obtained in the steps given by the PAV method where the last one contains the calibrated probabilities.

instance	initial score	PAV step 1	PAV step 2
e_1	0.76	0.765	0.765
e_2	0.77	0.765	0.765
e_3	0.70	0.705	0.705
e_4	0.71	0.705	0.705
e_5	0.66	0.685	0.686
e_6	0.71	0.685	0.686
e_7	0.69	0.69	0.686
e_8	0.68	0.68	0.68
e_9	0.48	0.485	0.485
e_{10}	0.49	0.485	0.485

Platt's Method

(Platt, 1999) presents a parametric approach for fitting a sigmoid that maps estimated probabilities into calibrated ones. This method was developed to transform the outputs of a support vector machine (SVM) from the original values $[-\infty, \infty]$ to probabilities, but can be extended to other types of models or probabilities variations. The idea consists on passing to the values a sigmoid function of the form:

$$p^*(i,j) = \frac{1}{1 + e^{A \times p(i,j) + B}}$$

The parameters A and B are determined such that minimise the negative log-likelihood of the data.

Platt's method is most effective when the distortion in the predicted probabilities has a sigmoid form (as in the previous example). Isotonic regression is more flexible and can be applied to any monotonic distortion. Nevertheless, isotonic regression used to present overfitting problems in some cases. Also, all the above methods can use the training set or an additional validation set for calibrating the model. The quality of the calibration might depend on this possibility and the size of the dataset. This is a recurrent issue in calibration, and it has been shown that some methods are better than others for small calibration sets (i.e. Platt's scaling is more effective than isotonic regression when the calibration set is small (Caruana & Niculescu-Mizil, 2004)).

Other Related Calibration Methods

Apart from the methods for obtaining calibrated probabilities, there exits other calibration techniques only applicable to specific learning methods. For instance, smoothing by m-estimate (Cestnik, 1990) and Laplace (Provost & Domingos, 2000) are another alternative ways of improving the probability estimates given by an unpruned decision tree. Probabilities are generated from decision trees as follows. Let *T* be a decision tree and *l* a leaf which contain *n* training

instances. If k of these instances are of one class (for instance, of positive class), then when T is applied to classify new examples it assigns a probability of $p = \frac{k}{n}$ that each example i in l belongs to the positive class. But using the frequencies derived from the count of instances of each class in a leaf might not give reliable probability estimates (for instance, if there are few instances in a leaf). So, for a two-class problem, the Laplace correction method replaces the probability estimate by $p' = \frac{k+1}{n+2}$. For a more general multiclass problem with C classes, the Laplace correction is calculated as $p' = \frac{k+1}{n+C}$. As Zadrozny and Elkan (2001) say "the Laplace correction method adjusts probability estimates to be closer to 1/2, which is not reasonable when the two classes are far from equiprobable, as is the case in many real-world applications. In general, one should consider the overall average probability of the positive class, i.e. the base rate, when smoothing probability estimates" (p. 610). Thus, smoothing by m-estimate consists in replacing the above mentioned probability estimate by $p' = \frac{k + b \cdot m}{n + m}$ where b is the base rate and m is the parameter for controlling the shift towards b. Given a base rate b, Zadrozny and Elkan (2001) suggest using m such that $b \cdot m = 10$.

Another related technique also applicable to decision trees is curtailment (Zadrozny & Elkan, 2001). The idea is to replace the score of a small leaf (i.e., a leaf with few training instances) by the estimate of its parent node, if it contains enough examples. If the parent node still have few examples, we proceed with its parent node and so on until to reach either a node sufficiently populated or the tree root.

CALIBRATION METHODS FOR RD

The regression case when the goal is to have that the expected output to be equal (or close) to the real average output (type RD), has been implicitly or explicitly considered in most regression techniques to date. This is so, because there are two numeric outputs (the predicted value and the real value), so, there is more variety of corrective functions to apply. First we are going to see which the problem is in this case. The idea can be depicted (see Figure 2) if we compare the behaviour of the test data, denoted by "real", and the model that we want to calibrate ("predicted"). In the figure, we show the behaviour of two models with a similar squared error.

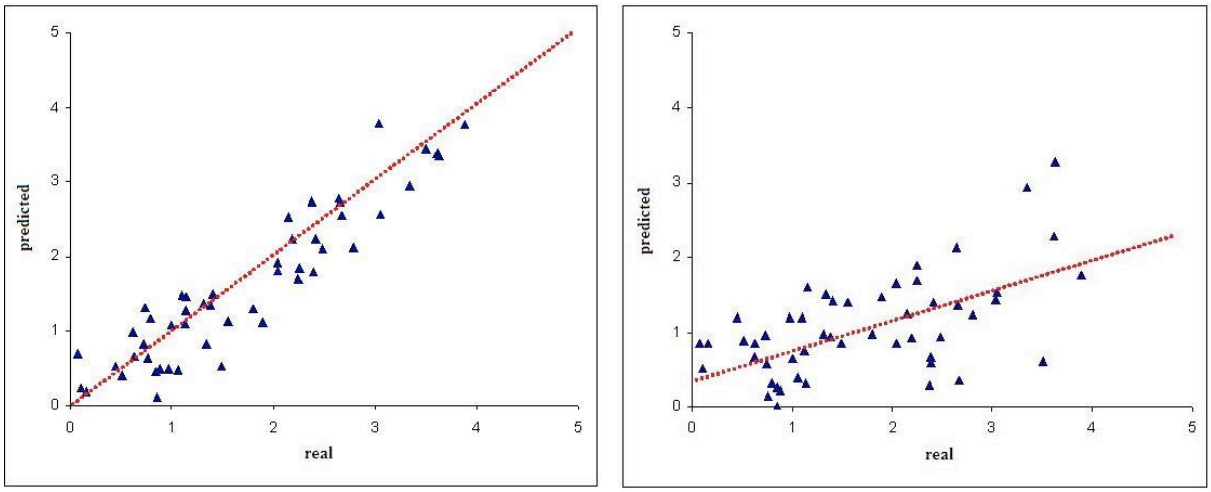

Figure 2. Calibration of regression models. Left: an uncalibrated model. Right: a calibrated model.

As we said in the introduction, the characteristic of a calibrated model for type RD is that the errors are equally distributed for the different output values, in other words, the expected value from distinct functions between the predicted value and the real value must be the right according to the function. For example, the expected value of the difference between the estimated value (y_{est}) and the real value (y) must be near to zero, so $E(y_{est} - y) = 0$. If this value is less than 0, then the real values are a little higher that the estimated ones, in average, if this value is greater than 0, the real values are a little lower than the estimated ones. Most regression models used to have this difference quite well calibrated. On contrary, the expected value of the quotient between the estimated value and the real value should be near to one, so $E(y_{est} / y) = 1$. If this quotient is greater than one, the error used to be positive for high values and negative for low values. Techniques such as linear regression use to give calibrated models, but others (nonlinear regression, local regression, neural networks, decision trees, etc.) can give uncalibrated models.

Logically, in both cases, the errors can be corrected, for example, by decreasing the high values and increasing the low values. In general, the solution comes from obtaining some type of estimation of the decalibration function between the real values and the predicted values. One of the most common approximations consists on calculate a linear regression as in the previous plots in the Figure 2 and apply it to the model, with the aim of fitting the calibration. These calibrations used to increase the mean squared error $\frac{(y - y_{est})^2}{n}$, but can achieve to reduce the relative mean squared error $\frac{(y - y_{est})^2}{(y - mean(y_{est})) \times n}$ or the error tolerance.

When the decalibration function is nonlinear (but it has a pattern), the problem of calibration becomes more complex, and some kind of nonlinear or local regression is needed to calibrate the model. In these cases, it is not properly a calibration process but a meta-learner, with several stages (stacking, cascading, etc.).

CALIBRATION METHODS FOR RP

On type RP, the prediction is a probability density function, and it is this function what we need to improve. This is a much more complex problem since improving this type of calibration can be done by mangling the prediction estimates (i.e. the MSE can be increased as the result of calibrating). Consequently, a trade-off must be found. In (Carney & Cunningham, 2006), they approach the problem by formulating it as a multi-objective problem. The two objectives of sharpness (a classical quality criterion based on negative log-likelihood (NLL) and the Anderson-Darling (A^2) test over the probability integral transform, as a measure of pure calibration.

FUTURE TRENDS

Future trends on calibration include a clearer recognition of the effects calibration has and when and how the four types of calibration are related as well as how they relate to classical quality metrics. In particular, calibration and traditional measures are sometimes conflicting, and we need to use a couple of metrics (such as A^2 and NLL in the case of type RP), or a hybrid one (such as MSE^p in the case of type CP), to select the best model.

In classification, most calibration methods we have analysed work for binary problems. As we have seen, most calibration methods are based on sorting the instances and/or making bins. But for more than two classes it is not so clear how to sort the instances or, more generally, how to make the bins. This is one of the reasons for which.the calibration problem has been less discussed. There are some works like (Zadrozny & Elkan, 2002) where the multiclass calibration problem has been studied,

but using approaches for reducing a multiclass problem to a set of binary problems and for finding an approximate solution for this problem.

Another interesting research line consists in to study in depth the relationship between ROC analysis (or its counterpart for regression, REC analysis) and calibration. For instance, the use of repairing concavities techniques in ROC analysis (Flach & Wu, 2005) to solve conflicts between the original class ranking and the new estimated probabilities.

Finally, type RP is a future trend on its own, since it is the most complex case which has been paid attention much more recently.

CONCLUSIONS

In this chapter we have addressed the problem of predictive model calibration and we have presented the most known calibration techniques for classification and regression.

We have shown that for classification, there are evaluation measures which are suitable for calibration (such as logloss, MSE, ...). And, also, there are another measures (for instance, accuracy) which are not good. Other measures are used in conjunction with calibration measures, especially the separability measures (AUC). Similarly, in regression, specific measures are needed to evaluate calibration, although must be usually accompanied by measures of sharpness.

Calibration techniques are usually based on deriving a transformation which converts the values (on types CD and RD) or the probabilities (on types CP and RP) to better estimates. Very different transformation techniques have been devised in the literature, but they usually include some kind of binning or sorting in discrete cases (classification) or some kind of integral in the continuous cases (regression).

REFERENCES

Ayer, M., Brunk, H., Ewing, G., Reid, W., & Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 5, 641–647.

Carney, M., & Cunningham, P. (2006). Making good probability estimates for regression. *17th European Conference on Machine Learning*, LNCS: Vol. 4212 (pp. 582-589).

Caruana, R., & Niculescu-Mizil, A.(2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp 69–78).

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. *Ninth European Conference on Artificial Intelligence* (pp. 147–149).

DeGroot, M. & Fienberg, S. (1982). The comparison and evaluation of forecasters. *Statistician*, 31(1), 12–22.

Dowe, D. L., Farr, G. E., Hurst, A. J., & Lentin, K. L. (1996). Information-theoretic football tipping. *3rd Conference on Maths and Computers in Sport* (pp 233-241).

Fawcett, T. (2006), An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.

Fawcett, T. & Niculescu-Mizil, A. (2007) PAV and the ROC convex hull. *Machine Learning*, 68(1), 97–106.

Flach, P. A., & Matsubara, E. T. (2007). A simple lexicographic ranker and probability estimator. *18th European Conference on Machine Learning* (pp. 575–582).

Flach, P.A., & Wu, S. (2005). Repairing concavities in ROC curves. *International Joint Conference on Artificial Intelligence (IJCAI'05)* (pp. 702–707).

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society*. Series B 14, 107-114.

Good, I. J. (1968). Corroboration, explanation, evolving probability, simplicity, and a sharpened razor. *British Journal of the Philosophy of Science*. 19, 123-143.

Lachiche, N., & Flach, P.A. (2003) Improving Accuracy and Cost of Two-class and Multi-class Probabilistic Classifiers Using ROC Curves. *International Conference on Machine Learning* (pp. 416-423).

Murphy, A. H. (1972). Scalar and vector partitions of the probability score: Part ii. n-state situation. *Journal of Applied Meteorology*, 11:1182–1192.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. J. Smola, editors, *Advances in Large Margin Classifiers* (pp. 61–74). MIT Press, Boston.

Provost, F., & Domingos, P. (2000). *Well-trained PETs: Improving probability estimation trees* (Technical Report CDER #00-04-IS). Stern School of Business, New York University.

Sanders, F. (1963). On subjective probability forecasting. *Journal of Applied Meteorology*, 2, 191-201.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *Eighteenth International Conference on Machine Learning* (pp. 609–616).

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 694–699).

KEY TERMS AND THEIR DEFINITIONS

Calibration technique: is any postprocessing technique which aims at improving the probability estimation or to improve error distribution of a given model.

Distribution calibration in classification (or simply "class calibration"): the degree of approximation of the true or empirical class distribution with the estimated class distribution.

Probabilistic calibration for classification: the degree of approximation of the predicted probabilities to the actual probabilities.

Distribution calibration in regression: the relation between the expected value of the estimated value and the mean of the real value must be unbiased at the global and local levels.

Probabilistic calibration for regression: when we have "density forecasting" models, a good calibration requires in general that these density functions are particular for each prediction, narrow when the prediction is confident and broader when it is less so.

Calibration measure: any kind of quality function which is able to assess the degree of calibration of a predictive model.

Confusion matrix: is a visual way of showing the recount of cases of the predicted classes and their actual values. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

Reliability diagrams. In these diagrams, the prediction space is discretised into 10 intervals (from 0 to 0.1, from 0.1 to 0.2, etc.). The examples whose probability is between 0 and 0.1 go into the first interval, the examples between 0.1 and 0.2 go into the second, etc. For each interval, the mean predicted value (in other words, the mean predicted probability) is plotted (x axis) in front of the fraction of positive real cases (y axis). If the model is calibrated the points will be near to the diagonal.

BIOGRAPHY

Antonio Bella finished his degree in Computer Science at the Technical University of Valencia in 2004 and started PhD studies in Machine Learning at the Department of Information System and Computation in the same university. At the same time, in 2005 he obtained a MSc in Corporative Networks and Systems Integration and in 2007 he started a degree in Statistical Science and Technology at the University of Valencia.

Cèsar Ferri is an associate professor of computer science at the Department of Information Systems and Computation, Technical University of Valencia, Spain, where he has been working since 1999. He obtained his BSc at the Technical University of Valencia, and his MSc at the University of Pisa, Italy. His research interests include machine learning, cost-sensitive learning, relational data mining, and declarative programming. He has published several journal articles, books, book chapters and conference papers on these topics.

José Hernández-Orallo, BSc, MSc (Computer Science, Technical University of Valencia, Spain), MSc (Computer Science, ENSEA, Paris), Ph.D. (Logic, University of Valencia). Since 1996, he has been with the Department of Information Systems and Computation, Technical University of Valencia, where he is currently an Associate Professor. His research interests centre on the areas of artificial intelligence, machine learning, data mining, data warehousing and software engineering, with several books, book chapters, journal and conference articles on these topics.

María José Ramírez-Quintana received the BSc from the University of Valencia (Spain) and the Msc and PhD in Computer Science from the Technical University of Valencia (Spain). She is currently an associate professor at the Department of Information Systems and Computation, Technical University of Valencia. She has lectured several courses on software engineering, functional and logic programming, and multiparadigm programming. Her research interest include mutiparadigm programming, machine learning, data mining algorithms, model combination and evaluation, and learning from structured data, with more than 60 publications in these areas, including journal articles, books, book chapters and conference contributions.