Similarity-Binning Averaging: A Generalisation of Binning Calibration

A. Bella, C. Ferri, J. Hernández-Orallo, and M.J. Ramírez-Quintana

Universidad Politécnica de Valencia, DSIC, Valencia, Spain

Abstract. In this paper we revisit the problem of classifier calibration, motivated by the issue that existing calibration methods ignore the problem attributes (i.e., they are univariate). These methods only use the estimated probability as input and ignore other important information, such as the original attributes of the problem. We propose a new calibration method inspired in binning-based methods in which the calibrated probabilities are obtained from k instances from a dataset. Bins are constructed by including the k-most similar instances, considering not only estimated probabilities but also the original attributes. This method has been experimentally evaluated wrt. two calibration measures, including a comparison with other traditional calibration methods. The results show that the new method outperforms the most commonly used calibration methods.

1 Introduction

Many machine learning techniques used for classification are good in discriminating classes but are poorer in estimating probabilities. One reason for this is that when many of these techniques were developed, the relevance and need of good probability estimates was not so clear as it is today. Another reason is that learning a good classifier (a model which tells accurately between two or more classes, i.e., with high accuracy) is usually easier than learning a good class probability estimator. In fact, in the last decade, there has been an enormous interest in improving classifier methods to obtain good rankers, since a good ranker is more difficult to obtain than a good classifier, but still simpler than a good class probability estimator. There are, of course, some other approaches which have addressed the general problem directly, i.e., some classification techniques have been developed and evaluated using probabilistic evaluation measures into account, such as the Minimum Squared Error (MSE) or LogLoss, in the quest for good class probability estimators.

In this context, and instead of redesigning any existing method to directly obtain good probabilities, some calibration techniques have been developed to date. A calibration technique is any postprocessing technique which aims at improving the probability estimation of a given classifier. Given a general calibration technique, we can use it to improve class probabilities of 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.

This work is motivated by the realisation that existing calibration methods only use the estimated probability to calculate the calibrated probability (i.e., they are univariate). In fact, most calibration methods are based on sorting the instances and/or making bins, such as binning averaging [10] or isotonic regression [3], where the only information which is used to sort or create these bins is the estimated probability. The same happens with other "mapping" methods, such as Platt's method [8]. However, more information is usually available for every instance, such as the original attributes of the problem.

In this paper, we introduce a new calibration method, called Similarity-Binning Averaging (SBA), which is similar to binning methods in the sense that the calibrated probability is calculated from k elements. Instead of sorting the examples in order to compute the bins, we use similarity to compute the k-most similar instances to conform one unique bin for each example. For that purpose, our approach uses not only the estimated probability but the instance attributes too. As a consequence, the resulting learned function is non-monotonic. That means that not only calibration will be affected, but discrimination will also be affected (and hence measures such as the Area Under the ROC Curve (AUC) or even qualitative measures such as accuracy).

The paper is organised as follows, in Section 2, some of the most-known calibration evaluation measures and calibration methods are reviewed. Next, Section 3 presents our calibration method based on binning. An experimental evaluation of the different calibration methods wrt. several measures is included in Section 4. Finally, Section 5 concludes the paper and points out the future work.

2 Calibration Methods and Evaluation Measures

In this section we review some of the most-known calibration methods and introduce the evaluation measures we will employ to estimate the calibration of a classifier. 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. $p(j)$ 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].

2.1 Calibration Methods

As we have mentioned in the introduction, the objective of calibration methods (as a postprocessing) is to transform the original estimated probabilities[1] Some well-known calibration methods are:

- The binning averaging method [10] consists in sorting the examples in decreasing order by their estimated probabilities and dividing the set into k

[1] Scores can also be used [11].

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.

- For two-class problems, [3] presented a pair-adjacent violators algorithm (PAV) for calculating the isotonic regression. The first step is to order decreasingly the n elements according to estimated probability and to initialise $p^*(i,j) = f(i,j)$. The idea is that calibrated probability estimates must be a monotone decreasing sequence, i.e., $p_1^* \geq p_2^* \geq \ldots \geq p_n^*$. 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. This process is repeated (using the new values) until an isotonic set is reached.
- Platt [8] presents a parametric approach for fitting a sigmoid that maps estimated probabilities into calibrated ones[2].

2.2 Calibration Evaluation Measures

Several measures have been proposed and used for evaluating the calibration of a classifier:

- Mean Squared Error (MSE) or Brier Score penalises strong deviations from the true probability.

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

  Although originally MSE is not a calibration measure, it was decomposed in [7] in terms of calibration loss and refinement loss.
- A calibration measure based on overlapping binning is CalBin [2]. 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 $\sum_{i^* \in 1..100} |p(i^*, j) - \hat{f}_j|$. 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 [2] suggests) is that it might be a much too large bin for small datasets. Instead of 100 we set a different bin length, $s = n/10$, to make it more size-independent.
- There exist other ways of measuring calibration such as Calibration Loss [5], chi-squared test through Hosmer-Lemeshow C-hat (decile bins) or H-hat (fixed thresholds), or through LogLoss.

For a more extensive survey of classification measures we refer the reader to [4].

[2] Originally, Platt proposed this method to be applied to SVM scores.

3 Calibration by Multivariate Similarity-Binning Averaging

As we have shown in the previous section, most calibration methods are based on a univariate transformation function over the original estimated class probability. In binning averaging, isotonic regression or Platt's method, this function is just obtained through very particular mapping methods, using $p(i, j)$ (the estimated probability) as the only input variable. Leaving out the rest of information of each instance (e.g., their original attributes) is a great waste of information which would be useful for the calibration process. For instance, in the case of binning-based methods, the bins are exclusively constructed by sorting the estimated probability of the elements. Binning can be modified in such a way that bins overlap or bins move as windows, but it still only depends on one variable (the estimated probability).

The core of our approach is to change the idea of "sorting" for creating bins, into the idea of using similarity to create bins which are specific for each instance. The rationale for this idea is as follows. If bins are created by using only the estimated probability, calibrated probabilities will be computed from possibly different examples with similar probabilities. The effect of calibration is small, since we average similar probabilities. On the contrary, if we construct the bins using similar examples according to other features, probabilities can be more diverse and calibration will have more effect. Additionally, it will be sensitive to strong probability deviation given by small changes in one or more original features. This means that if noise on a variable dramatically affects the output, probabilities will be smoothed and, hence, they will be more noise-tolerant. For instance, if $(3, 2, a)$ has class $true$ and $(2, 2, a)$ has class $false$, the estimated probability for $(3, 2, a)$ should not be too close to 1.

Based on this reasoning, we have implemented a new calibration method: Similarity-Binning Averaging (SBA). In this method the original attributes and the estimated probability are used to calculate the calibrated one.

We have split the method into 3 stages. The "Stage 0" is the typical learning process. A classification technique is applied to a training dataset to learn a probabilistic classification model (M). In the training dataset, X_{ij} are the attributes and Y_i is the class. This stage may not exist if the model is given beforehand (a hand-made model or an old model).

On the left of figure 1 we can observe the "Stage 1" of the SBA method. In this stage, the trained model M gives the estimated probabilities associated with a dataset. This dataset can be the same used for training, or an additional validation dataset VD, as we have shown in figure 1. The estimated probabilities $p(i, j)$ $1 \leq j \leq c$ are joined as new attributes for each instance i of VD, creating a new dataset VDP. Finally, on the right of figure 1 shows the "Stage 2" of the SBA method. To calibrate a new instance I, first, the estimated probabilities are estimated from the classification model M, and these probabilities are added to the instance creating a new instance (IP). Next, the k-most similar instances to this new instance are selected from the dataset VDP. Finally, the calibrated probability of this instance I for each class j is the predicted class probability

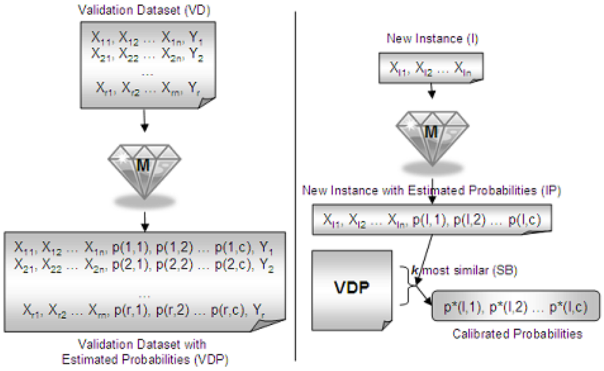
Validation Dataset (VD)
X11, X12 ... X1n, Y1
X21, X22 ... X2n, Y2
...
Xr1, Xr2 ... Xrn, Yr
M
X11, X12 ... X1n, p(1,1), p(1,2) ... p(1,c), Y1
X21, X22 ... X2n, p(2,1), p(2,2) ... p(2,c), Y2
...
Xr1, Xr2 ... Xrn, p(r,1), p(r,2) ... p(r,c), Yr
Validation Dataset with Estimated Probabilities (VDP)
New Instance (I)
XI1, XI2 ... XIn
M
New Instance with Estimated Probabilities (IP)
XI1, XI2 ... XIn, p(I,1), p(I,2) ... p(I,c)
VDP
k most similar (SB)
p*(I,1), p*(I,2) ... p*(I,c)
Calibrated Probabilities

Fig. 1. Left: Stage 1 of the SBA method. Right: Stage 2 of the SBA method.

of the k-most similar instances using their attributes.

Our method is similar to "cascading" [6]. The main difference is that in our method the calibrated probability is the predicted class probability of the k-most similar instances using their attributes and class, i.e., in the first stage instead of adding the class to the instance (as cascading would do), the estimated probabilities of each class are added.

4 Experimental Results

For the experimental evaluation, we have implemented the evaluation measures and the calibration methods explained at Sections 2 and 3, and we have used machine learning algorithms implemented in the data mining suite WEKA [9].

Initially, we have selected 20 (small and medium-sized) binary datasets (table 1) from the UCI repository [1]. We evaluate the methods in two different settings: training and test set, and training, validation and test set. The reason is because the calibration methods can use or not an additional dataset to calibrate. In the training/test setting, (we add a "T" at the end of the name of the methods) randomly, each dataset is split into two different subsets: the training and the test sets (75% and 25% of the instances, respectively). In the training/validation/test setting, (we add a "V" at the end of the name of the methods) randomly, each dataset is split into three different subsets: the training, the validation and the test sets (56%, 19% and 25% of the instances, respectively). Four different methods for classification have been used (with their default parameters in WEKA): NaiveBayes, J48 (a C4.5 implementation), IBk ($k = 10$) (a k-NN implementation) and Logistic (a logistic regression implementation). A total of 400 repetitions have been performed for each dataset (100 with each classifier). In each repetition, for the training/test setting, the training set is used to train a classifier and calibrate the probabilities of the model, and the test set is used to test the calibration of the model, while for the training/validation/test setting, the training set is used to train a classifier, the validation set is used to calibrate the probabilities of the model, and the test set is used to test the calibration of the model. Furthermore, in each repetition the same training, validation and test sets are used for all methods.

The calibration methods used in the experiments are: binning averaging (with 10 bins), PAV algorithm, Platt's method, and Similarity-Binning Aver-

#	Datasets	Size	Nom.	Num.	#	Datasets	Size	Nom.	Num.
1	Breast Cancer	286	9	0	11	House Voting	435	16	0
2	Wisconsin Breast Cancer	699	0	9	12	Ionosphere	351	0	34
3	Chess	3196	36	0	13	Labor	57	8	8
4	Horse Colic	368	15	7	14	Monks1	556	6	0
5	Credit Rating	690	9	6	15	Mushroom	8124	22	0
6	German Credit	1000	13	7	16	Sick	3772	22	7
7	Pima Diabetes	768	0	8	17	Sonar	208	0	60
8	Haberman BreastW	306	0	3	18	Spam	4601	0	57
9	Heart Statlog	270	0	13	19	Spect	80	0	44
10	Hepatitis	155	13	6	20	Tic-tac	958	8	0

Table 1. Datasets used in the experiments. Size and number of nominal and numeric attributes.

aging (SBA) (with $k = 10$). All of them have been evaluated for the CalBin and MSE calibration measures. Apart from comparing the results of the calibration methods, we also compare them with two reference methods:

- Base: is the value obtained with the classification techniques without calibration.
- 10-NN[3]: is the value of using the 10 most similar instances (just with the original attributes) to directly estimate the calibrated probability. This method is just to show the importance of using the estimated probabilities as inputs to compute the similarity.

In tables 2 and 3 we show the results with respect to CalBin and MSE measures for each method (for both measures the lower the better). These values are the average of the 400 repetitions for each dataset.

	ClassT	10-NNT	BinT	PAVT	PlattT	SBAT	BinV	PAVV	PlattV	SBAV
1	0.1953	0.1431	0.2280	0.2321	0.1856	0.1827	0.3092	0.2928	0.2446	0.1924
2	0.0494	0.0374	0.0647	0.0447	0.0623	0.0408	0.0791	0.0538	0.0775	0.0423
3	0.0698	0.1472	0.0501	0.0397	0.0434	0.0491	0.0548	0.0448	0.0479	0.0628
4	0.1517	0.1216	0.1533	0.1535	0.1421	0.1164	0.1996	0.1853	0.1563	0.1244
5	0.1220	0.0882	0.1060	0.1035	0.1132	0.0848	0.1408	0.1293	0.1269	0.0874
6	0.1250	0.1340	0.1263	0.1393	0.1268	0.1233	0.1933	0.1855	0.1227	0.1233
7	0.1192	0.1049	0.1220	0.1351	0.1205	0.1105	0.1889	0.1861	0.1267	0.1199
8	0.1984	0.2028	0.2316	0.2400	0.1998	0.1994	0.2877	0.2798	0.2777	0.2149
9	0.1476	0.1412	0.1690	0.1587	0.1529	0.1443	0.2247	0.1995	0.1834	0.1432
10	0.1632	0.1359	0.1643	0.1673	0.1727	0.1332	0.2082	0.1999	0.2597	0.1358
11	0.0665	0.0625	0.0777	0.0542	0.0791	0.0516	0.0945	0.0672	0.1006	0.0588
12	0.1380	0.1588	0.1179	0.0990	0.1358	0.1064	0.1701	0.1428	0.1854	0.1303
13	0.1876	0.2996	0.1984	0.1464	0.2110	0.1820	0.2914	0.1940	0.4478	0.2792
14	0.1442	0.2794	0.1618	0.1355	0.1046	0.1443	0.2067	0.1740	0.1340	0.1730
15	0.0395	0.0366	0.0418	0.0358	0.0468	0.0368	0.0434	0.0359	0.0494	0.0367
16	0.0296	0.0158	0.0270	0.0236	0.0250	0.0194	0.0297	0.0264	0.0285	0.0265
17	0.2606	0.1916	0.2343	0.2376	0.2374	0.2007	0.3207	0.2924	0.2750	0.1844
18	0.0945	0.0471	0.0636	0.0568	0.0951	0.0466	0.0733	0.0658	0.0964	0.0910
19	0.3138	0.3497	0.2995	0.2911	0.3110	0.3110	0.3615	0.3380	0.4265	0.3117
20	0.1240	0.2094	0.1260	0.1198	0.0906	0.0934	0.1736	0.1621	0.0971	0.0824
AVG.	0.1370	0.1453	0.1382	0.1307	0.1328	**0.1188**	0.1826	0.1628	0.1732	0.1310

Table 2. Results by dataset: Measure CalBin. Training/test setting (T) and Training/validation/test setting (V).

As we can see in the last row of tables 2 and 3, with both calibration measures our method SBA with the training/test setting has obtained the best results and our method SBA with the training/validation/test setting has obtained good results as well.

There are some differences between the results when calibration methods are evaluated with each measure (CalBin and MSE) (tables 2 and 3). These differences come from the different nature of the measures. While CalBin is a measure

[3] Implemented by an IBk with $k = 10$ in WEKA

	ClassT	10-NNT	BinT	PAVT	PlattT	SBAT	BinV	PAVV	PlattV	SBAV
1	0.2086	0.1912	0.2086	0.2095	0.2016	0.1998	0.2123	0.2136	0.2081	0.1982
2	0.0353	0.0262	0.0510	0.0343	0.0362	0.0306	0.0635	0.0375	0.0380	0.0316
3	0.0465	0.0648	0.0467	0.0387	0.0391	0.0227	0.0515	0.0424	0.0423	0.0332
4	0.1506	0.1351	0.1503	0.1449	0.1442	0.1336	0.1641	0.1535	0.1513	0.1371
5	0.1347	0.1121	0.1244	0.1203	0.1262	0.1160	0.1320	0.1239	0.1287	0.1176
6	0.1889	0.1795	0.1888	0.1883	0.1877	0.1814	0.1849	0.1832	0.1821	0.1800
7	0.1790	0.1749	0.1795	0.1786	0.1777	0.1821	0.1818	0.1779	0.1756	0.1835
8	0.1926	0.1992	0.1924	0.1936	0.1906	0.2005	0.1966	0.1982	0.1974	0.1957
9	0.1491	0.1435	0.1580	0.1503	0.1470	0.1469	0.1675	0.1534	0.1522	0.1390
10	0.1473	0.1294	0.1460	0.1483	0.1397	0.1305	0.1546	0.1505	0.1585	0.1271
11	0.0554	0.0568	0.0646	0.0482	0.0538	0.0456	0.0816	0.0574	0.0599	0.0524
12	0.1266	0.1297	0.1118	0.0981	0.1094	0.0996	0.1311	0.1071	0.1186	0.1141
13	0.1120	0.1233	0.1582	0.1128	0.1044	0.0907	0.2109	0.1419	0.2288	0.1196
14	0.1214	0.1047	0.1172	0.1030	0.1065	0.0517	0.1362	0.1164	0.1244	0.0819
15	0.0083	0.0006	0.0174	0.0040	0.0079	0.0001	0.0198	0.0045	0.0088	0.0003
16	0.0310	0.0311	0.0355	0.0266	0.0307	0.0241	0.0370	0.0275	0.0277	0.0370
17	0.2545	0.1760	0.2343	0.2286	0.2285	0.2080	0.2305	0.2157	0.2225	0.1847
18	0.1027	0.0814	0.0765	0.0721	0.0878	0.0690	0.0800	0.0746	0.0895	0.1042
19	0.2829	0.2459	0.2776	0.2637	0.2437	0.2692	0.2639	0.2482	0.2568	0.2276
20	0.1579	0.1141	0.1480	0.1448	0.1468	0.0817	0.1571	0.1522	0.1526	0.1108
AVG.	0.1343	0.1210	0.1343	0.1254	0.1255	**0.1142**	0.1428	0.1290	0.1362	**0.1188**

Table 3. Results by dataset: Measure MSE. Training/test setting (T) and Training/validation/test setting (V).

that only evaluates calibration, MSE also evaluates other components.

It is important to remark that we are making general comparisons between methods in equal conditions. First of all we are comparing to classification methods without calibration (Base). Logically, calibration would not have had any sense if we had not improved the results. The second comparison is with the 10-NN method, which is related to our method, but only uses the original attributes of the problem to make the bin of the 10 elements which are more similar and to obtain the calibrated probability. The other three methods we compare to only use the estimated probability to calculate the calibrated probability. The most interesting comparison is with the binning averaging method, because our method is also based on the idea of binning.

If we observe table 3, it is important to remark how our method improves significantly the other calibration methods in terms of MSE.

Additional experiments: grouped by classification method, suitable statistical tests to confirm the differences in the results are significant and multiclass experiments can be found at: http://users.dsic.upv.es/∼abella/MulticlassExperiments.pdf.

5 Conclusions

In this work we have revisited the problem of class probability calibration. We have generalised the idea of binning by constructing the bins using similarity to select the k-most similar instances. In this way, we have a different bin for each example and, of course, bins can overlap. Similarity is not computed by only using the estimated probabilities but also with the problem attributes. Our hypothesis was that calibration would be more effective the more information we are able to provide for computing this similarity. Leaving the problem attributes out (as traditional calibration methods do) is like making the problem harder than it is.

The implementation of the method is straightforward through any off-the-shelf k-NN algorithm. Consequently, our method is closely related to the cascade generalisation method [6].

The experimental results we have presented here confirm the previous hypothesis and show a significant increase in calibration for the two calibration measures considered, over three well-known and baseline calibration techniques: non-overlapping binning averaging, Platt's method and PAV. It is true that this calibration is partly obtained because of the increase of AUC and it is, consequently, non monotonic, but in many applications where calibration is necessary the restriction of being monotonic is not only applicable, but it is an inconvenience. In fact, when calibrating a model, the original model and class assignments can be preserved, while the only thing that has to be modified is the new probabilities. In other words, the predictions of a comprehensible model composed of a dozen rules can be annotated by the estimated probabilities while preserving the comprehensibility of the original model.

As future work we are working on attribute-weighted k-NN to form the bins in order to gauge the importance of attributes for cases when there is a great number of attributes or a great number of classes. Similarly, we want to use locally-weighted k-NN, where closer examples have more weight, in order to make the method more independent from k.

Another future work is the analysis of the method for multiclass problems, because as we have seen in the definition of our method, it can be applied to multiclass problems. We have to compare with other approximations like [11] and find some other calibration methods, since binning, Platt's and PAV cannot deal directly with multiclass problems.

References

1. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
2. R. Caruana and A. Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proc. of the 10th Intl. Conference on Knowledge Discovery and Data Mining*, pages 69–78, 2004.
3. M. Ayer et al. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 5:641–647, 1955.
4. C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recogn. Lett.*, 30(1):27–38, 2009.
5. P. Flach and E. Matsubara. A simple lexicographic ranker and probability estimator. In *18th European Conference on Machine Learning*, pages 575–582, 2007.
6. J. Gama and P. Brazdil. Cascade generalization. *Machine Learning*, 41:315–343, 2000.
7. A. H. Murphy. Scalar and vector partitions of the probability score: Part ii. n-state situation. *Journal of Applied Meteorology*, 11:1182–1192, 1972.
8. J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, Boston, 1999.
9. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Elsevier, 2005.
10. B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proc. of the 18th Intl. Conference on Machine Learning*, pages 609–616, 2001.
11. B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *The 8th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining*, pages 694–699. ACM, 2002.