Quantification via Probability Estimators

Antonio Bella*, Cèsar Ferri*, José Hernández-Orallo* and María José Ramírez-Quintana*
*DSIC-ELP, Universidad Politécnica de Valencia, Camí de Vera s/n, 46022 Valencia, Spain
E-mails: {abella, cferri, jorallo, mramirez}@dsic.upv.es

***Abstract*—Quantification is the name given to a novel machine learning task which deals with correctly estimating the number of elements of one class in a set of examples. The output of a quantifier is a real value; since training instances are the same as a classification problem, a natural approach is to train a classifier and to derive a quantifier from it. Some previous works have shown that just classifying the instances and counting the examples belonging to the class of interest (*classify & count*) typically yields bad quantifiers, especially when the class distribution may vary between training and test. Hence, adjusted versions of *classify & count* have been developed by using modified thresholds. However, previous works have explicitly discarded (without a deep analysis) any possible approach based on the probability estimations of the classifier. In this paper, we present a method based on averaging the probability estimations of a classifier with a very simple scaling that does perform reasonably well, showing that probability estimators for quantification capture a richer view of the problem than methods based on a threshold.**

***Keywords*-quantification; probability estimators; class imbalance; classification;**

I. Introduction

George Forman [1][2][3] has introduced and systematised a new supervised machine learning task called 'quantification'. Quantification is defined as follows: "given a labeled training set, induce a *quantifier* that takes an unlabeled test set as input and returns its best estimate of the class distribution."[2]. For instance, consider a bank that has a credit risk assessment model (possibly a machine learning classifier), and it is assigned a new portfolio of customers (e.g., 100,000 new customers who originated from an agreement with a retailing company). A very important (and classical) question is to determine which customers the credits will be granted to. However, before resolving all of these specific decision making problems, the bank will typically require an assessment of how many credits it will grant, i.e., the bank will need to quantify how many of the customers in the portfolio will have their credit approved. The accuracy of this quantification is critical to assigning human and economical resources, long before any specific decision is made. Also, the result of this estimation may affect the thresholds that may be established for each particular operation since the assessment of the global risk can affect the way in which each local risk is managed (e.g. the policies will probably change if the plan was to work with 25,000 positive results but we expect 50,000). The quantification problem can be found in almost any area in data mining, such as failure detection, medical diagnosis, customer-relationship management, and retailing.

The task is closely related to classification since examples have the same presentation (several input features and a nominal output feature), but it is different in that we are not interested in the specific predictions for each instance of an application dataset, but on the overall count of elements of a particular class. Consequently, quantification is applied to a *batch* of examples, and not to a single example alone. Since the output of the quantification problem is a real value, it has a relation to regression, but the input of the regressor would be a single example rather than a set of examples.

Quantification is a very frequent problem in real applications, so it is somewhat surprising that nobody in the data mining community, until George Forman, recognised, baptised, and addressed this task on its own. In [1][2][3], Forman develops several methods and defines new experimental settings to evaluate the task, especially focussing on the cases where the training class distribution is different to the test class distribution.

One of the first conclusions from these works is that the naïve solution called *classify & count (CC)* does not work well. Consequently, several other methods are introduced, by properly adjusting the threshold and also scaling the result. Other methods are not based on *CC* but are still based on thresholds. However, there is another way to address the problem, which is to consider probability estimators instead of crisp classifiers. If a classifier estimates a class probability for each example, the *CC* method becomes the *probability estimation & average (P&A)* method. Surprisingly, this approach is considered a "non-solution" in Section 3.4 in [1] and an "ill-conceived method" in Section 2.4 in [3]. The reason for this is clearly shown with an example: "For a well-calibrated classifier, the output value $y = 70\%$ indicates that, of the training cases that score similarly, approximately 70% were positive. Supposing a large test set contains the same proportion of positives, as among those cases that score $y = 70\% \pm \epsilon$, roughly 70% would be positive. However, if we repeat the test with most of the negative cases removed at random, then the proportion of positives among the remaining test cases scoring in this bin

This work has been partially supported by the EU (FEDER) and the Spanish MEC/MICINN, under grant TIN 2007-68093-C02, the Spanish project "Agreement Technologies" (Consolider Ingenio CSD2007-00022) and the GVA project PROMETEO/2008/051.

($y = 70\% \pm \epsilon$) would be much greater. Thus, the output of 70% would greatly underestimate the new $P(+|y = 70\%)$; and likewise for every bin. Again, the end effect is that this method would underestimate at high prevalence and overestimate at low prevalence, just like the *CC* method. As a result, the use of probability estimators has not been explored since it has been considered a "non-solution".

We agree with the rationale, but this does not necessarily imply that we should not give adjusted versions of *P&A* a chance. If adjusted versions of *CC* work, such as Forman's AC and $T50$, we think we could explore similar (or different) adjusting methods for *P&A*. In this paper, we present a simple scaling method over *P&A*, called "Scaled Probability Average", which shows very good performance. A quantifier based on probability estimation is not based on a threshold, so the appraisal of this threshold is not so critical. In fact, our method depends on the quality of all the probabilities since it considers all the information that the classifier gives us about the dataset.

Summing up, the main contributions of this paper are that we introduce probability estimation to the scene of quantification, and we derive a simple scaling method that shows good performance.

The paper is organised as follows. Section II introduces basic notation and terminology for the quantification problem and sets the quality measures for quantification. Section III derives the methods based on *P&A*, presents the plain version *Probability Average* (which is actually a bad solution in general) and then introduces the *Scaled Probability Average* method. These two methods are evaluated in Section IV with a range of test class imbalances and then compared to other quantification methods. Finally, Section V wraps up the paper with the conclusions and future work.

II. Notation and Previous Work

Given a dataset T, n denotes the number of examples, and c the number of classes. We will use i to index or refer to examples, so we will express $i = 1 \dots n$ or $i \in T$ indistinctly. $f(i, j)$ 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 not strictly a probability but a single-label indicator function. With $n_j = \sum_{i=1}^{n} f(i, j)$, we denote the number of examples of class j. $\pi(j)$ denotes the prior probability of class j, i.e., $\pi(j) = n_j/n$. When referring to a particular dataset T, we will use the equivalent expression $\pi_T(j) = \frac{\sum_{i \in T} f(i,j)}{|T|}$. Given a classifier, $p(i, j)$ represents the estimated probability of example i to be of class j taking values in [0,1]. $\hat{\pi}(j)$ denotes the estimated probability of class j which is defined as $\hat{\pi}_T(j) = \frac{\sum_{i \in T} p(i,j)}{|T|}$ when referring to a dataset T. For the sake of readability, when $c = 2$, we will use the symbols $\oplus$ for the positive class and $\ominus$ for the negative class. Since the probabilities are complementary for two classes, we will focus on the positive class. $C_\theta(i, j)$ is 1 iff j is the predicted class obtained from $p(i, j)$ using a threshold θ. We can omit θ when it is embedded in the classifier or clear from the context. When $c = 2$, we will use the following measures $TP = \sum_{i=1}^{n} f(i, \oplus)C(i, \oplus)$, $TN = \sum_{i=1}^{n} f(i, \ominus)C(i, \ominus)$, $FP = \sum_{i=1}^{n} f(i, \ominus)C(i, \oplus)$, $FN = \sum_{i=1}^{n} f(i, \oplus)C(i, \ominus)$; we will also use the ratios, $tpr = TP/(TP + FN)$ and $fpr = FP/(FP + TN)$. We will use pos for the actual proportion of positives, i.e., $\pi(\oplus)$; and we will use neg for the actual proportion of negatives, i.e., $\pi(\ominus)$. Finally, the function $clip(X, a, b)$ truncates a real value X inside an interval $[a, b]$. We represent the elements of T of class $\oplus$ and $\ominus$ with $T_\oplus$ and $T_\ominus$, respectively.

A. Quantification

Forman [1][2][3] was the first to identify and name the quantification problem: "quantification is accurately estimating the number of positives in the test dataset as opposed to classifying individual cases accurately." Therefore, the main objective in a quantification task is to estimate the class distribution in the target population from the distribution observed in the training dataset. This problem is especially important in many application areas, such as medicine, risk assessment, diagnosis, etc., where the training dataset does not represent a random sample of the target population (because population changes over time or the classes are highly imbalanced, with the positive class as a minority).

As Forman pointed out, quantification is a machine learning task that is quite different from classification. In quantification, we are interested in the test set as a whole in order to determine its class distributions and not in the individual predictions for each example. In fact, although accurate classifications generally give accurate class counting, an inaccurate classifier can also be a good quantifier if false positives and false negatives cancel each other. In [3], Forman introduced several quantification methods that we arrange into the following three groups:

- Methods based on counting the positive predicted examples. The *classify & count* (CC) and the *adjusted count* (AC) methods belong to this group.
- Methods based on selecting a classifier threshold, but in this case, the threshold is determined from the relationship between tpr and fpr in order to provide better quantifier estimates. For instance, some methods choose one particular threshold, such as: the *X method*, which selects the threshold that satisfies $fpr = 1 - tpr$; the *Max method*, which selects the threshold that maximises the difference $tpr - fpr$; or those methods like $T50$ that select a particular rate between tpr and fpr. The *Median Sweep (MS) method*[1] is another method that tests all the thresholds in the test set, estimates the

[1]It is proposed when the tpr and fpr are estimated from cross-validation.

number of positives in each one, and returns a mean or median of these estimations.

- Methods based on a mixture of distributions. The *Mixture Model* (MM)[1] is included in this group. It consists of determining the distributions from the classifier scores on the training positive ($D_{\oplus}$) and negative examples ($D_{\ominus}$) and then modelling the observed distribution D on the test set as a mixture of $D_{\oplus}$ and $D_{\ominus}$.

The best results are obtained with AC in general; however, when the training sets are small and the number of positives is small, other methods such as $T50$ or MS can get better results (at the cost of performing worse in other more balanced situations). Of the above-mentioned methods proposed by Forman, the simplest one is the CC method. It consists of learning a classifier from the training dataset and counting the examples of the test set that the classifier predicts positive ($\sum_{i \in Test} C(i, \oplus)$). This method gives poor results since it underestimates/overestimates the proportion of positives, unless the classifier is perfect ($tpr = 1$ and $fpr = 0$). The AC method is an improvement of the CC method, which estimates the true proportion of positives $\widehat{pos}$ by applying the equation $\widehat{pos} = \frac{\widehat{pos}' - fpr}{tpr - fpr}$, where $\widehat{pos}'$ is the proportion of predicted positives $\frac{\sum_{i \in Test} C(i,\oplus)}{|Test|}$. Forman proposed estimating tpr and fpr by cross-validation on the training set. Since this scaling can give negative results or results above 1, the last step is to clip $\widehat{pos}$ to the range $[0..1]$. Finally, $T50$ is another method that selects the threshold where $tpr = 50\%$ and the rest works the same as AC. This method supposedly behaves better when the denominator in the formula of $\widehat{pos}$ is unstable with AC. We have implemented these three methods in order to fairly compare the performance of our proposals with them. The reason for this choice is that these methods are the ones that are most related to ours and their performance is good, or very good, with respect to other quantification methods such as X, Max, or MS. In other words, they are representative of the state-of-the-art in quantification. Motivated by the fact that in quantification only one result per dataset is produced, Forman proposed an experimental methodology to evaluate quantification that is different from the one that is normally used for classification. It consists in varying the class distribution between the training set and the test set. For the training set, Forman randomly selected the number of positive and negative examples from a range. For the test set, he varied the percentage of positive examples to also cover scenarios with imbalanced classes.

As we have mentioned in the introduction, a natural approach to solve the quantification problem that Forman disregarded is to use a probability estimator instead of a classifier. But, we will show that there is a different way to estimate the positive proportion by using probability estimations sharing the spirit of the CC method.

Moreover, our proposal is supposed to be more robust to variations in the probability estimation of few examples than other methods based on thresholds because we take into account all the information from the dataset. For example, consider a test set with probabilities and actual classes as follows: (0.90,+), (0.55,+), (0.53,−), (0.51,−) and (0.21,−). If we set the threshold at 0.6, the proportion of positives is 20%; however, if the threshold is 0.5, then the proportion of positives is 80%. Note that this is a good classifier, with perfect ordering, i.e., AUC=1. Therefore, methods that use thresholds are less robust in the sense that a change in the estimation of few examples could cause a small shift in the threshold but a large variation in the positive count estimation. However, regarding the probabilities of all the examples, the proportion of positives is 54%. In order to have a large change here, many probability estimations would have to change significantly.

B. Quantification Evaluation

We use two global evaluation measures from the classical error measures for continuous variables, the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) for quantification. Forman only used the absolute error in his experiments, but we consider that the MSE measure is a better way to quantify differences between estimations for real values. Let us formalise these measures in order to better establish the quantification problem and goal.

Consider that we have a method that estimates the proportion of elements for each class ($\hat{\pi}_T(j)$). By calculating the absolute difference of these two values, we have the global MAE for each class, $GMAE_j(T) = |\pi_T(j) - \hat{\pi}_T(j)|$, and for all the classes we have $GMAE(T) = \frac{1}{c} \cdot \sum_{j=1..c} GMAE_j(T)$. Similarly, we calculate $GMSE_j(T)\ (\pi_T(j) - \hat{\pi}_T(j))^2$ and $GMSE(T) = \frac{1}{c} \cdot \sum_{j=1..c} GMSE_j(T)$. For binary problems, we have that $GMAE_{\oplus} = GMAE_{\ominus} = GMAE$ and also that $GMSE_{\oplus} = GMSE_{\ominus} = GMSE$. Therefore, for binary problems, we will only evaluate the error for the proportion of positives.

III. Quantifying by Scaled Averaged Probabilities

The idea of using an average of the probability estimations is supported by the issue that probabilities represent much richer information than just the decisions, which are simply derived information from the probability estimation using a threshold. After this rationale, the use of probabilities shapes a family of methods that we call *probability estimation & average*. The simplest method in this family is called *Probability Average* (PA). First, a probabilistic classifier is learned from the training data, such as a Probability Estimation Tree or a Naïve Bayes model. Then, the learned model is applied to the instances in the test set, obtaining a probability estimation for each one. Finally, the average

of the estimated probabilities for each class is calculated. Although this definition is multiclass, for the rest of the paper we will concentrate on binary datasets. In this case, we only need to care about one class (the positive class), and the method is defined as follows: $\hat{\pi}^{PA}_{Test}(\oplus) = \frac{\sum_{i \in Test} p(i,\oplus)}{|Test|}$.

Logically, if the proportion of positive examples in the training set is different from the proportion of positive examples in the test set, the result will not be satisfactory in general. The solution comes precisely from the analysis of the extreme case when all the elements in the test set are of one class. In this case, we will get the average probability for the positive cases alone, which can only be 1 for a perfect classifier (which is not frequently the case). As in the AC method, the idea is to use a proper scaling.

Nevertheless, from the training set, it is possible to calculate the *actual proportion of positive examples* ($\pi_{Train}(\oplus)$), the *positive probability average* ($\hat{\pi}_{Train}(\oplus)$), the *positive probability average for the positives* ($\hat{\pi}_{Train_\oplus}(\oplus)$), and the *positive probability average for the negatives* ($\hat{\pi}_{Train_\ominus}(\oplus)$).

From the definitions, it is easy to check the following: $\hat{\pi}_{Train_\oplus}(\oplus) \cdot \pi_{Train}(\oplus) + \hat{\pi}_{Train_\ominus}(\oplus) \cdot (1 - \pi_{Train}(\oplus)) = \hat{\pi}_{Train}(\oplus)$.

From this equation, we derive $\pi_{Train}(\oplus) = \frac{\hat{\pi}_{Train}(\oplus) - \hat{\pi}_{Train_\ominus}(\oplus)}{\hat{\pi}_{Train_\oplus}(\oplus) - \hat{\pi}_{Train_\ominus}(\oplus)}$, which yields a probabilistic version of Forman's adjustment (see Fig.1). When all are positives, $\hat{\pi}_{Train_\oplus}(\oplus)$ sets the maximum, and we scale this to 1. When all are negatives, $\hat{\pi}_{Train_\ominus}(\oplus)$ sets the minimum, and we scale this to 0.

Thus, we propose a new quantification method, which we call *Scaled Probability Average* (SPA), applying this last formula (*scaling*) in the same way as Forman to the value obtained with the PA method ($\hat{\pi}^{PA}_{Test}(\oplus)$), i.e., $\hat{\pi}^{SPA}_{Test}(\oplus) = \frac{\hat{\pi}^{PA}_{Test}(\oplus) - \hat{\pi}_{Train_\ominus}(\oplus)}{\hat{\pi}_{Train_\oplus}(\oplus) - \hat{\pi}_{Train_\ominus}(\oplus)}$.

$$\hat{\pi}^{SCC}_{Test}(\oplus) = \frac{\frac{\sum_{i \in Test} C(i,\oplus)}{|Test|} - \hat{\pi}_{Train_\ominus}(\oplus)}{\hat{\pi}_{Train_\oplus}(\oplus) - \hat{\pi}_{Train_\ominus}(\oplus)}$$

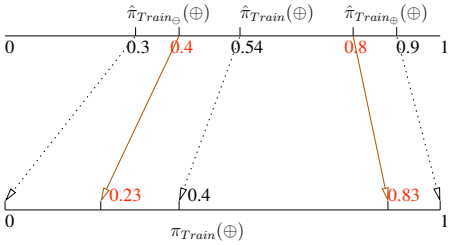

Figure 1. Scaling used in the SPA method. The limits in the training set are placed at 0.3 and 0.9. The estimated value for the training set is 0.54 whereas the actual proportion in the training set is 0.4. The scaling would move a case at 0.4 to 0.23 and a case at 0.8 to 0.83.

In the same way as in the SPA method, the proportion of positive examples estimated by the PA method is scaled. This scaling can also be applied to the proportion of positive examples estimated by the CC method. Therefore, we propose the *Scaled Classify & Count* (SCC) method.

IV. Experiments

In this section, we present an experimental evaluation of several quantification methods: those based on classification (i.e., on a threshold), namely CC, AC, T50 (explained in Section II); our two methods based on probability average, namely PA and SPA; and a hybrid method (SCC), which is explained in Section III. We have not used an internal cross-validation process with the training set to better estimate the thresholds, tpr and fpr, as Forman does. Instead, we have used an additional validation set to estimate the thresholds, tpr and fpr. In other words, the methods are the same, but instead of using a 50 fold cross-validation process with the training set, we have used two different sets (training and validation) and repeated the process 100 times.

The experimental setting is based on the common case where we train a classifier or probability estimator on a training dataset and we want to quantify the number of examples of one class for a different test dataset. We are especially interested in cases where class distributions vary between training and test. In order to get this variation (and being able to consider cases where the majority class examples are increased or reduced), for each problem, we divided the whole dataset into 37.5% for training, 37.5% for validation, and 25% for test. With this original test set, we constructed six different variations test sets: the whole test set with the original proportion of classes, and five test sets changing the proportion of classes (100% of examples of positive class and 0% of negative class, 75% and 25%, 50% and 50%, 25% and 75%, and 0% and 100%). The proportions were obtained by random undersampling using a uniform distribution.

In order to have a broad range of classifiers and probability estimators, we used four different methods from WEKA [4]: NaïveBayes, J48 (a C4.5 implementation), IBk ($k = 10$) (a k-NN implementation), and Logistic (a logistic regression implementation). We used the WEKA default parameters in the methods (except the parameter k in the IBk method that is set to 10). A total of 100 repetitions were performed for each dataset (25 for each classifier). We selected 20 binary datasets (Table I) from the UCI repository [5]. We adopted the same criterion as Forman, and the positive class is the minority class.

In Table I, we show the results with respect to the GMSE measure[2] for each method. These values are the average of the 100 repetitions for each dataset. We also include MSE, AUC, CalBin and Accuracy measures (see, e.g. [6] for a definition of these measures). For each dataset, as suggested in [7], we calculated a Friedman test and showed that the six methods do not have identical effects, so we calculated the Nemenyi post-hoc test to compare all the methods with each other (with a probability of 99.5%). The values in bold show that this method outperforms the others and that the

[2]GMAE results portray a similar picture.

Table I
GMSE MEASURE FOR EACH DATASET AND THE AVERAGE RESULT OF ALL THE DATASETS FOR EACH QUANTIFICATION METHOD, AND MSE, AUC, CALBIN AND ACCURACY MEASURES FOR EACH DATASET.

#	Datasets	Size	$\pi_{Tr}(\oplus)$	CC	AC	T50	SCC	PA	SPA	MSE	AUC	CalBin	Acc.
1	Wisconsin Breast Cancer	699	0.34	0.0022	0.0008	0.0667	0.0008	0.0032	0.0006	0.04	0.82	0.04	0.95
2	Chess	3196	0.48	0.0033	0.0002	0.0155	0.0024	0.0116	0.0001	0.06	0.82	0.09	0.93
3	Credit Rating	690	0.45	0.0219	0.0046	0.0115	0.0070	0.0286	0.0037	0.15	0.76	0.16	0.82
4	German Credit	1000	0.31	0.0921	0.0265	0.0277	0.0598	0.0878	**0.0176**	0.24	0.65	0.26	0.65
5	Pima Diabetes	768	0.35	0.0572	0.0160	0.0247	0.0332	0.0642	0.0099	0.21	0.69	0.23	0.70
6	House Voting	435	0.38	0.0044	0.0018	0.0522	0.0018	0.0048	0.0014	0.07	0.81	0.07	0.92
7	Monks1	556	0.49	0.0449	0.0072	0.0186	0.1025	0.0487	0.0057	0.16	0.72	0.23	0.79
8	Mushroom	8124	0.48	0.0268	0.0004	0.0380	0.0022	0.0211	0.0003	0.12	0.80	0.13	0.84
9	Spam	4601	0.39	0.0118	0.0003	0.0284	0.0006	0.0156	0.0003	0.10	0.80	0.10	0.88
10	Tic-tac	958	0.34	0.0626	0.0107	0.0127	0.0525	0.0671	**0.0062**	0.19	0.71	0.23	0.72
11	Breast Cancer	286	0.29	0.1280	0.1118	0.1171	0.1638	0.1057	**0.0829**	0.28	0.60	0.31	0.62
12	Haberman Breast	306	0.27	0.2045	0.1478	0.1265	0.2555	0.1316	**0.1049**	0.28	0.59	0.31	0.60
13	Heart Disease	303	0.45	0.0236	0.0108	0.0235	0.0116	0.0291	0.0083	0.15	0.75	0.17	0.80
14	Heart Statlog	270	0.43	0.0251	0.0141	0.0352	0.0152	0.0320	0.0111	0.15	0.75	0.18	0.80
15	Ionosphere	351	0.35	0.0429	0.0075	0.0446	0.0068	0.0487	0.0053	0.17	0.75	0.19	0.81
16	Monks2	601	0.34	0.2386	0.1826	0.2146	0.3087	**0.1168**	0.1732	0.27	0.55	0.31	0.52
17	Monks3	554	0.48	0.0010	0.0004	0.0151	0.0069	0.0151	0.0006	0.05	0.82	0.13	0.97
18	Hepatitis	155	0.19	0.1536	0.1241	0.1329	0.1454	0.1226	**0.0885**	0.27	0.66	0.30	0.66
19	Sonar	208	0.46	0.0525	0.0460	0.0622	0.0521	0.0572	0.0400	0.26	0.67	0.27	0.69
20	Spect	80	0.49	0.1009	0.0921	0.1308	0.1530	0.0722	0.0879	0.24	0.68	0.27	0.70
	AVG.		0.39	0.0649	0.0403	0.0599	0.0691	0.0542	0.0324	0.17	0.72	0.2	0.77

difference is statistically significant. Also, if several values are underlined, this indicates that these methods outperform the rest with a statistically significant difference, but the difference between the two is not significant.

The results are categorical as far as the use of probability estimators. Not only do they work, but they outperform all the other methods. Of course, as we expected, a proper scaling is the key to success. In the same way that AC dramatically improves CC, SPA also improves PA. However, by looking at the table in more detail we can get more insight. For instance, while CC and PA are relatively close, SPA is able to improve PA more than AC is able to improve CC (in relative terms). The reason may be found in the way that SPA is not based on thresholds (or measures such as tpr or fpr which depend on them), but on averages. Thus, the results show that SPA is more robust. Regarding the other methods, SCC is a hybrid method that uses classification to estimate the base proportion and uses the probability average to scale it. The result shows no clear improvement over CC or PA. This supports the thesis that, for SPA, it is the conjunction of PA with the scaling what makes the results good. The behaviour of $T50$ is expected, since it highly depends on the threshold and this is a method oriented to cases where the original training set is highly imbalanced.

In order to analyse the effect of imbalance (in the training set), we repeated the experiments but reduced the number of examples of the positive class by 90% in the training and validation sets. In this situation, the SPA method still obtains good results, but, in most cases, the differences between the SPA method and the AC and $T50$ methods are not statistically significant. We want to remark that AC and $T50$ methods are designed for situations where the proportion of classes are imbalanced. Even though, this second experimental setting is more favourable for the AC and $T50$ methods, SPA method still obtains good results. These results are summarised in Fig.2 (right).

Finally, in order to analyse the effect with respect to test imbalance, for each dataset, we analysed the results for the six different test set imbalances. The conclusion was that the behaviour is similar in all the datasets. The best results (as expected) are obtained with class proportions that are close to the original class proportion or close to 0.5, and much worse as we get closer to 0 or 1 proportions. In Fig.2 (left), we show a typical case of a dataset where the errors are much lower when the test proportion is close to the original training proportion. In Fig.2 (centre), we show the average results for the 20 datasets without reducing the proportion of examples of the positive class in the training set. In Fig.2 (right), we show the results of reducing the proportion of examples of the positive class by 90% in the training and validation sets. Fig.2 (centre) shows that when the proportion of examples of the positive class in the test set is more or less between 20% and 60%, the AC method is the one that obtains the best results. In the rest of proportions, the SPA is the best method because its behaviour is always quite similar. With this in mind, it is logical to think in terms of a new "mixed" method, which uses the method that obtain the best results in each interval. We have not implemented this method, but we propose its implementation and experimental evaluation as an interesting topic for future work.

V. CONCLUSIONS AND FUTURE WORK

Quantification is an old and frequent problem that has only very recently received proper attention as a separate machine learning task. In our opinion, one of the most natural ways to address the problem is to average the probability estimations for each class because we use all

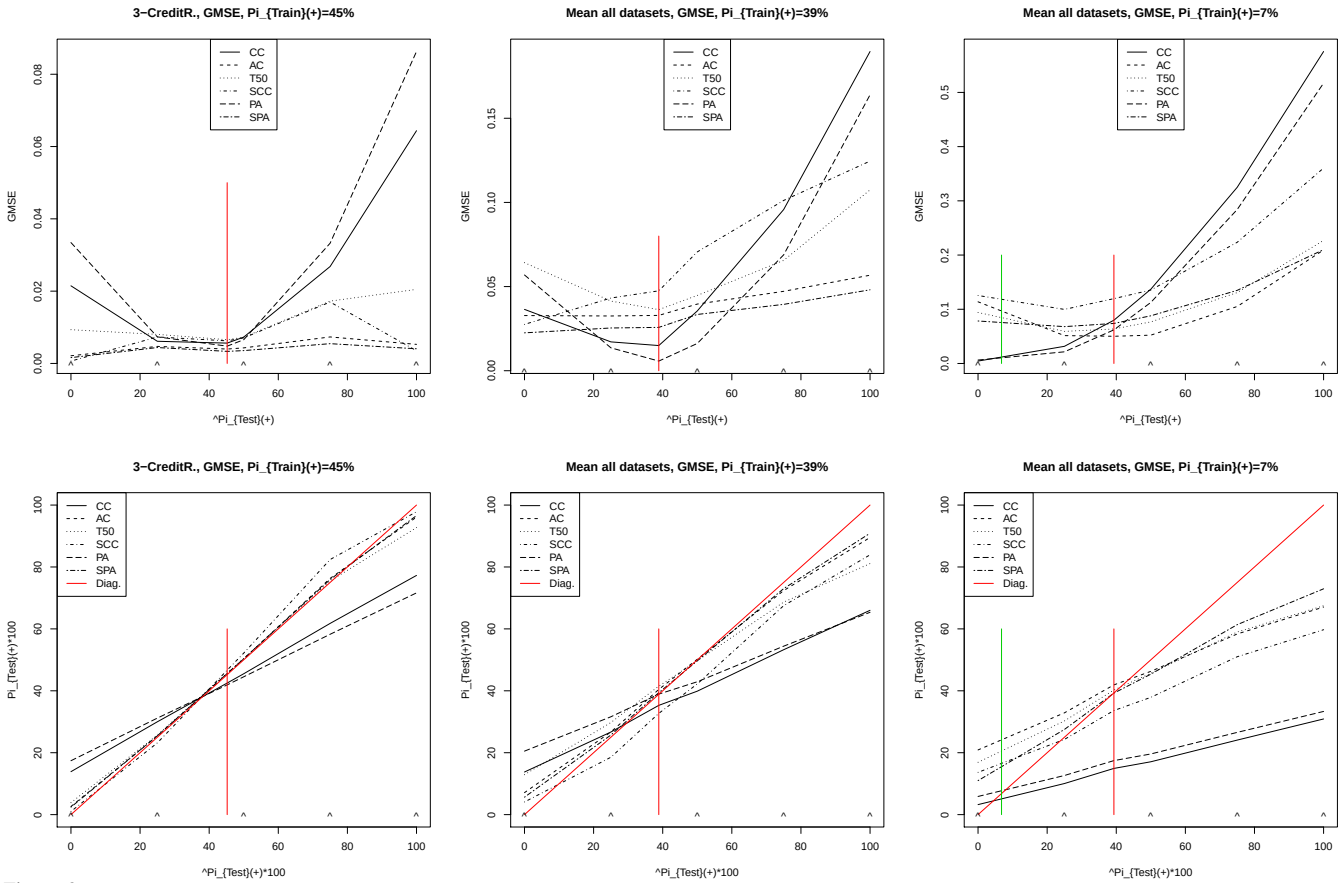

Figure 2. Upper-Left: GMSE results with variable test positive ratios for the dataset CreditR. Upper-Centre: GMSE results with variable test positive ratios for the average of all the datasets. Upper-Right: GMSE results with variable test positive ratios for the average of all the datasets, with the proportion of positive examples in the training and validation sets reduced by 90%. Lower-(Left, Centre and Right): The value of each quantification method for each test setting. The diagonal shows the perfect value.

the information provided by the probability estimator and because we do not rely so much on a good threshold choice. However, in the same way that *CC* does not work, this easy approach does not work. Since *CC* can be adjusted, we have seen that a simple probability average method can also be adjusted. We have derived a generalisation of Forman's scaling for probabilities, and we have derived a new method from it. The results are highly positive and show that the use of probability estimators for quantification is good to pursue, which can lead to new methods in the future.

As further future work, there are obviously several areas (cost quantification [3]) and experimental settings (very imbalanced training datasets) that are borrowed from Forman's seminal works for which a more exhaustive evaluation of our methods and the derivation of specialised ones should be addressed. One of these extensions is quantification for more than two classes. Our PA method is originally multiclass, but the scaling used in SPA is not. We think that proper methods (different to indirect 1vs1 or 1vsall) could be derived using probability averaging.

Finally, quantification for regression is also a machine learning task that should be investigated (as we stated in the introduction with the credit assessment problem, where an estimation of the total amount for a customer portfolio and not the number of granted credits would be the goal). In fact, quantification for regression might be closer to quantification through probability averaging than through classification.

References

[1] G. Forman, "Counting positives accurately despite inaccurate classification," in *ECML*, 2005, pp. 564–575.

[2] ——, "Quantifying trends accurately despite classifier error and class imbalance," in *KDD*, 2006, pp. 157–166.

[3] ——, "Quantifying counts and costs via classification," *Data Min. Knowl. Discov.*, vol. 17, no. 2, pp. 164–206, 2008.

[4] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Elsevier, 2005.

[5] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998.

[6] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recogn. Lett.*, vol. 30, no. 1, pp. 27–38, 2009.

[7] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, January 2006.