Bella Sanjuán, A.; Ferri Ramírez, C.; Hernández Orallo, J.; Ramírez Quintana, MJ. (2014). Aggregative quantification for regression. Data Mining and Knowledge Discovery. 28(2):475-518. doi:10.1007/s10618-013-0308-z.

Noname manuscript No.
(will be inserted by the editor)

Aggregative quantification for regression

Antonio Bella · Cèsar Ferri · José Hernández-Orallo · María José Ramírez-Quintana

Received: date / Accepted: date

Abstract The problem of estimating the class distribution (or prevalence) for a new unlabelled dataset (from a possibly different distribution) is a very common problem which has been addressed in one way or another in the past decades. This problem has been recently reconsidered as a new task in data mining, renamed *quantification* when the estimation is performed as an *aggregation* (and possible adjustment) of a single-instance supervised model (e.g., a classifier). However, the study of quantification has been limited to classification, while it is clear that this problem also appears, perhaps even more frequently, with other predictive problems, such asregression. In this case, the goal is to determine a distribution or an aggregated indicator of the output variable for a new unlabelled dataset. In this paper, we introduce a comprehensive new taxonomy of quantification tasks, distinguishing between the estimation of the whole distribution and the estimation of some indicators (summary statistics), for both classification and regression. This distinction is especially useful for regression, since predictions are numerical values that can be aggregated in many different ways, as in multi-dimensional hierarchical data warehouses. We focus on aggregative quantification for regression and see that the approaches borrowed from classification do not work. We present several techniques based on segmentation which are able to produce accurate estimations of the expected value and the distribution of the output variable. We show experimentally that these methods especially excel for the relevant scenarios where training and test distributions dramatically differ.

Antonio Bella · Cèsar Ferri ·
José Hernández-Orallo · María José Ramírez-Quintana
DSIC-ELP, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain
Tel.: (+34) 96 387 73 50
Fax. (+34) 96 387 73 59
E-mail: abella@dsic.upv.es, cferri@dsic.upv.es, jorallo@dsic.upv.es, mramirez@dsic.upv.es

Keywords Quantification · Regression Quantification · Probability estimation · Segmentation · Distribution · Aggregation

1 Introduction

A common situation in data mining applications involves training a regression model predicting the expenditure, consumption, number of complaints, or any other numeric value y. For instance, imagine that we have learnt a model for individual customer expenditure from a customer portfolio X_1 (i.e., a dataset) that corresponds to a specific region, business area and time period, as extracted from a hierarchical multidimensional data warehouse. Eventually, we may want to apply the model to a *different* customer portfolio X_2, e.g., a different slice of the datamart for which we do not have the true value for y (i.e., X_2 is an unlabelled dataset). In order to assess this second portfolio, some typical questions might be: (1) "what's the expected average expenditure for the new portfolio?", (2) "what's the percentage of customers that will have an expenditure lower than 20 euros?", (3) "what's the typical expenditure for this portfolio?"

An answer to these questions can be given by just applying the regression model to each customer in the new portfolio X_2, so leading to a set $\hat{Y}_2$, which makes up a continuous empirical distribution $\hat{p}$. With this distribution, the above questions are just expressed as (1) $\mathbb{E}_{\hat{p}}[y]$ (i.e., the mean of $\hat{Y}_2$), (2) $Pr(y \leq 20 | y \in \hat{Y}_2)$ (i.e., a tail of the distribution) and (3) the value t for which $Pr(y \leq t | y \in \hat{Y}_2) = 0.5$ (i.e., the median).

The above example refers to the aggregation of a regression model, but the notion can also be applied to the aggregation of a classification model. In fact, the latter has received much more attention in the literature under different names, with the term class prevalence (or distribution) estimation being the most common (Neyman, 1938; Tenenbein, 1970; Alonzo et al, 2003). Many of these works focussed on how a small sample could be used to estimate the distribution of a bigger sample ('double' or 'two-phase' sampling) and not necessarily when the distributions change. Also, some of these works do not rely on a supervised model issuing an output value for every single instance in the dataset.

Forman (2005; 2006; 2008) presented the problem as a new supervised machine learning task called 'quantification'. Quantification (for classification) was 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"(Forman, 2006). Quantification is characterised (and distinguished from other distribution estimation problems) by how the problem is presented.

First, quantification focusses on cases where the training and test distributions differ (a distribution shift), because otherwise the quantification problem would be pointless: if the training and test distributions are equal, the best estimation for the test set seems to be the observed empirical distribution on

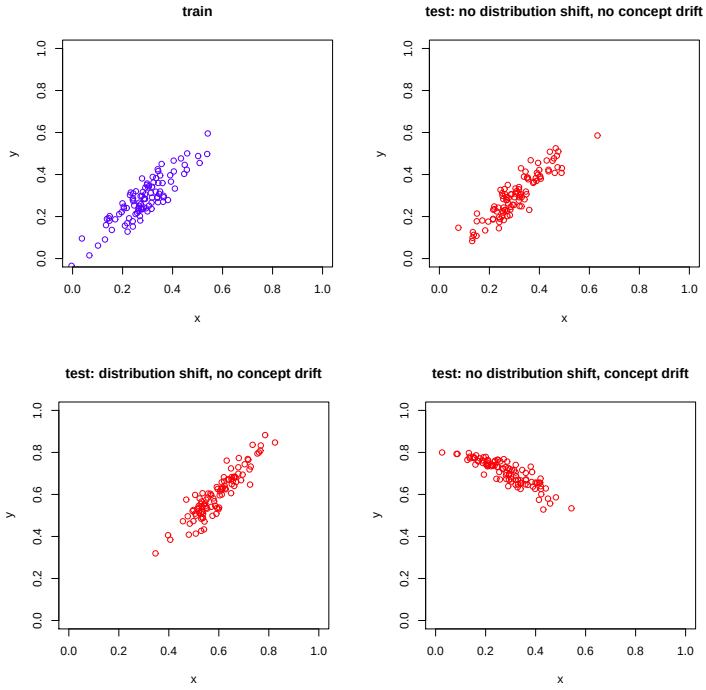

Fig. 1 An artificial example showing the applicability of quantification depending on the existence of (data) distribution shift and concept drift. Top left: actual distribution for the first batch (train). Top right: actual distribution for a second batch (test) where neither the concept nor the distribution have changed ($p(y|x)$ and $p(x)$ have not changed). While the estimation of the distribution may still be a problem when the number of training data is small, quantification is unnecessary here because other (and simpler) statistical approaches exist. Bottom left: actual distribution for a third batch (test) where the distribution ($p(x)$) has changed but the concept is the same ($p(y|x)$ has not changed), so an estimator $\hat{p}(y|x)$, trained on the first batch can still be useful. Quantification focusses on this problem. Bottom right: actual distribution for a fourth batch (test) where the distribution $p(x)$ has not changed but the concept has drifted ($p(y|x)$ has changed completely). Estimating the distribution here is a much more difficult problem.

the training data, possibly using a smoothing approach or other statistical techniques. We then expect a change in the distribution. This may be a covariate shift or a prior probability shift (Moreno-Torres et al, 2012), depending on whether the change originates on the covariates X or on the outputs Y. We will just use the term distribution shift for both cases. Importantly, a distribution shift is different from a concept drift, where the very target function changes between training and test. Figure 1 shows this difference.

Second, the problem is presented with a labelled training dataset from which we can learn a supervised model or estimate other parameters, which are then used to estimate the distribution for the whole unlabelled dataset —available as a *batch* of examples.

A different question is how the problem is solved. A general and practical approach is performed by aggregating the predictions of an underlying su-

pervised model (a regression model, a classification model, or a conditional probability estimator $\hat{p}(y|x)$), which gives a prediction for each single instance x. In other words, quantification is performed by *aggregating* the estimations for individual examples, with a possible adjustment of this aggregation, as we will see. This is especially appropriate in data mining applications where we already have (validated) predictive models, as well as applications with hierarchical data, where we train the model at the lower level and want to aggregate (*roll-up*) its predictions upwards, as usual in modern data warehouses. This differs from cases where we may have a global estimator or other individual, but unsupervised, estimators, such as a likelihood estimator $\hat{p}(x|y)$ or a joint distribution estimator $\hat{p}(x, y)$. This choice of single-instance supervised models is motivated by two observations: (1) predictive models are by far more common in data mining practice than likelihood or joint distribution estimators, and (2) the target function $p(y|x)$ will be the same since we do not consider concept drift, while $p(x|y)$ or $p(x, y)$ are different whenever the distribution changes. This means that a supervised model estimating $\hat{p}(y|x)$ can be preserved (differently from $\hat{p}(x|y)$ or $\hat{p}(x, y)$).

Given this setting for quantification, the simplest approach is then to calculate the predictions for all the examples using the model and aggregate them. This is, in fact, an ideal solution when the supervised model is perfect. However, many models are imperfect and biased (because of the difficulty of the problem: overfitting, underfitting, lack of data for some regions, and other factors). In fact, Forman showed that, for classification quantification, the naive method "classify & count" does not generally produce a good approximation of the actual distribution for the dependent value y. In other words, a biased predictive model may lead to bad estimations of the overall distribution of the dependent value, especially when this distribution is significantly different from the distribution used for training.

Interestingly, however, some not-so-good models can be unbiased, and aggregating their predictions may lead to good quantification, or there might be some quantification techniques for biased models that could reduce or correct their bias and lead to good quantification. Forman summarises this (Forman, 2008): "it is sufficient but not necessary to have a perfect classifier in order to estimate the class distribution well".

Since then, and given the large number of applications of quantification, new methods have been introduced that have improved the results for classification quantification, such as an "adjusted classify & count" (Forman, 2008), "median sweep" (Forman, 2008), and many others (Sánchez et al, 2008; Xue and Weiss, 2009; Bella et al, 2010; González-Castro et al, 2012), either by using crisp classifiers, or soft classifiers (rankers or probability estimators).

The problem of quantification for regression may have the same large number of applications as quantification for classification. Regression quantification addresses a very common situation: the prediction of aggregated numerical values such as sales, consumptions, duration, people, etc. However, to our knowledge, it has not been addressed in the context of distribution shift and using a base regression model.

While in classification this distribution is a discrete distribution (described as a set of probabilities for each class), in regression we can estimate a complete (empirical) continuous distribution and not only an expected mean. This also makes aggregative quantification for regression a more difficult task, as we will see, since the base regression models not only have bias on the location (the mean) but also have a tendency to compact the data and reduce the variability and dispersion of the output variable.

The goal of this paper is to develop new methods for regression quantification that can be applied over any predictive model built with off-the-shelf data mining software tools. As focussing on regression quantification, we will assume that the underlying predictive model is a regression model and we will concentrate most of our effort to this *direct* approach. Also, and just for comparison, we will briefly explore the indirect approach of using *classification* techniques applied to a discretised version of the problem.

We first explore the adaptation of several ideas from previous quantification methods, such as an "adjusted regress & sum". However, as we will see, this adaptation produces poor results. A better analysis of the problem leads to a novel approach based on the idea of segmentation. Instead of estimating the whole distribution (which may have many different shapes depending on the application) and use this to correct the error of the regression model, we just use a more flexible approach. We segment the training distribution into bins and use the errors in each bin to adjust (scale) the regression model. Using these segmentation techniques and simple adjustments for location and spread we are able to get much better results than the "adjusted regress & sum" method.

Our contribution in this paper is then manifold. Firstly, we give a more solid and comprehensive view of the quantification problem for several tasks, leading to a taxonomy of quantification approaches with their corresponding evaluation metrics. This taxonomy also distinguishes the cases where we want to estimate an indicator (e.g., summarised statistic) of the distribution or the whole distribution. Secondly, we show that the problem of regression quantification is richer than the problem of classification quantification, because we move from a discrete output to a continuous output. The ideas which work for classification quantification do not work for regression quantification (such as global adjustment), using a direct approach, unless we convert the regression problem to a classification problem through discretisation as an indirect approach. And thirdly, we propose new methods based on segmentation which are able to show good results even in the difficult distribution shift scenario we set in the experiments.

The paper is organised as follows. Section 2 introduces some notation, two examples and some previous work. From here we introduce a comprehensive taxonomy and a set of metrics for each quantification task in section 3. Then we focus on regression quantification in section 4, which analyses the problem more formally. We introduce several methods that are inspired by classification quantification and some new methods based on segmentation, adjustment and spread, all assuming underlying regression techniques. In contrast, some

other indirect methods based on a discretisation of the problem and the use of classification quantification are also defined for reference. Section 5 performs a thorough experimental evaluation of these methods for the indicator estimation case and the distribution estimation case. Finally, section 6 closes the paper with a discussion of results and some future work.

2 Background

In this section, we will introduce some notation to express what quantification is precisely. The understanding of this problem will be helped by two examples and a proper account of related work, including a short description of the methods which have been previously introduced for (classification) quantification.

2.1 Notation

We will deal with supervised (or predictive) problems, where the input and output domains are denoted by X and Y respectively. An unlabelled dataset is any subset (actually a multiset) of X. A labelled dataset D is any subset of $X \times Y$. We will use the terms D_X and D_Y for the projections of D for the input and output domains respectively. Occasionally, we will drop the subindex when clear from the context. Given an unlabelled or labelled dataset D of size $n = |D|$, we will assume a (strict) order such that we can just refer to an example with its index i in this order. Somewhat abusing notation we will express $i = 1 \dots n$ or $i \in D$ indistinctly. For the ith example, y_i will denote the true output value corresponding to the input value x_i. In this paper we refer to both classification and regression problems. In classification, the output domain is a set of nominal values $Y = \{l_1, l_2, \dots, l_c\}$ usually referred to as class labels or simply classes; whereas in regression problems the output values are real numbers ($Y \subset \mathbb{R}$). A crisp model is any function $m : X \to Y$. The estimation (or prediction) for input x_i is denoted by $\hat{y}_i$. A soft or probabilistic model is any function which returns a probability distribution for any given input value x, i.e., a conditional probability estimator $\hat{p}(y|x)$. For classification this is a categorical distribution and for regression this can be any continuous distribution. Typically, *Train* will denote the training dataset, while *Test* will denote the test dataset. In that follows, Pr means probability, p denotes a probability density function o discrete probability distribution function and P denotes a cumulative distribution function.

The (true) empirical (marginal) distributions for dataset D are given by the function $p_{D_X}(x)$ for the input values and $p_{D_Y}(y)$ for the output values.

In classification,

$$p_{D_Y}(l) \triangleq Pr(y = l | y \in D_Y) \tag{1}$$

is a categorical probability distribution (which gives a probability or frequency for each class label l). In the binary case, $\oplus$ will denote the positive class and $\ominus$ the negative class. For instance, given a binary dataset D with a 80% of class $\oplus$ then $p_{D_Y}(\oplus) = 0.8$.

In regression, $p_{D_Y}(r)$ is a probability density function for each real value r, with cumulative distribution function $(P_{D_Y}(r))$:

$$P_{D_Y}(r) \triangleq Pr(y \leq r | y \in D_Y) = \int_{-\infty}^{r} p_{D_Y}(y)dy \tag{2}$$

The expected value for this distribution is just the mean of D_Y, which is denoted by:

$$\mu_{D_Y} \triangleq \mathbb{E}[D_Y] = \int_{-\infty}^{\infty} y\, p_{D_Y}(y)\, dy = \frac{\sum_{i=1}^{n} y_i}{|D|} \tag{3}$$

And σ_{D_Y} denotes the standard deviation of the target values of D,

$$\sigma_{D_Y} \triangleq \sqrt{\frac{\sum_{i=1}^{n}(y_i - \mu_{D_Y})^2}{|D|^2}}$$

Given an unlabelled dataset D_X, we do not know p_{D_Y}. *Estimating this probability distribution is precisely what this paper is about.* Let us give a definition of quantification:

Definition 1 Quantification: Given a labelled training dataset $Train \subset X \times Y$ for a supervised problem $p(y|x)$, and given an unlabelled test dataset $Test$, the quantification problem is the estimation of p_{Test_Y} from $Train$. If Y is a discrete set, then we have a *classification quantification* problem, and p_{Test_Y} is a discrete (categorical) distribution. If Y is a continuous set, then we have a *regression quantification* problem, and p_{Test_Y} is a continuous distribution.

If this estimation is performed by aggregating the individual predictions $\hat{y}_i$ of a predictive model then we have an aggregative quantification approach. The trivial solution for the aggregative quantification problem for classification is defined as:

$$\hat{p}_{Test_Y}(l) \triangleq \frac{\sum_{i=1}^{n} I(\hat{y}_i = l)}{|Test|} \tag{4}$$

where I is the indicator function ($I(true) = 1$ and $I(false) = 0$). This solution is known as *Classify & Count* (CC). Similarly, the trivial solution for the quantification problem for regression is defined by the cumulative empirical distribution:

$$\hat{P}_{Test_Y}(r) \triangleq \frac{\sum_{i=1}^{n} I(\hat{y}_i \leq r)}{|Test|} \tag{5}$$

which can be called *Regress & Splice* (RS). Note that this gives a value for each possible r, which determines an estimated distribution. In quantification,

Fig. 2 A schematic view of how much information (and effort) we require depending on the problem we want to solve. This gradation is illustrated by the arrows which become implications when we have perfect estimations.

we are interested in the whole distribution of $Test_Y$ or some of its indicators, such as a mean or a median. Of course, this distribution or indicators can be well estimated by the use (e.g., aggregation) of very accurate individual predictions. However, it is important to realise again that we could also achieve good results from not-so-good individual predictions, provided they are not biased (locally and globally). As a thought experiment, consider that we scramble the predictions of a good regression model for an unlabelled dataset by just swapping an indefinite high number of predictions. After this, the distribution is exactly the same, even though the regression model becomes awful for individual predictions.

This unidirectional relation is shown in Figure 2, where we illustrate that individual predictions are much more informative and require more effort than the estimation of the whole distribution. This is illustrated by the arrows, which show some kind of summarisation. Only when we have *perfect* estimations, these arrows become implications. Interestingly, as we go from left to right less information and effort is required. As a result, this schema also suggests that we do not always need to derive the indicators from the distribution, or the distribution from the individual predictions. In fact, on some occasions, it may be better to estimate the indicators directly.

One single and generally useful indicator that can be calculated from this estimated distribution (Eq. 5) is its expected value, which is an estimation for μ_{Test_Y} above (Eq. 3),

$$\hat{\mu}_{Test_Y} \triangleq \frac{\sum_{i=1}^{n} \hat{y}_i}{|Test|} \tag{6}$$

which could be similarly called *Regress & Sum* (RS)[1]. In that follows, since we will focus on the output domain and distribution, we will usually drop Y in $Test_Y$.

2.2 Examples: understanding quantification

Given the notation above we will see two specific examples that will help to better understand what quantification is, and how it works when aggregating predictions from a base classification or regression model. We will also informally discuss some classical concepts that play an important role here, such as dataset imbalance or unevenness, overfitting, bias and variance, which will all be more formally addressed in section 4.1. Let us see an example for classification first:

[1] We use the same acronym for *Regress & Splice* and *Regress & Sum*, since both just aggregate the individual values with any further processing.

Example 1 A quantification problem in classification
Consider a car renting company which assesses the suitability (acceptability) of a car (unacceptable, acceptable, good or very good) according to several characteristics.[2] A classification model has been trained from data collected over a recent batch of cars which were supplied by the usual provider. Now, a deal is being negotiated with a new provider, which has given detailed information about the characteristics of all the cars in the new batch. The car renting company may be interested in calculating how many cars will be unacceptable. This is a quantification problem that can be solved by aggregating the predictions of the classification model for this new batch of cars. Figure 3 (top) shows the class distribution for the first batch ($Train$, left) and second batch ($Test$, right), which is not known by the car renting company and it is what we want to estimate. We see an important class distribution shift between $Train$ and $Test$. In this case, we approximate the test distribution with a decision tree learnt from the training dataset. Its confusion matrix for $Train$ and $Test$ is shown in the middle row of Figure 3. If we apply the decision tree to all the examples in the test set and plot the predicted class frequencies, we get the histogram on Figure 3 (bottom, right). As we can see, this estimated distribution significantly differs from the actual one. The estimation for class 'good' is almost perfect but a considerable error appears on 'unacc' and 'vgood'. Noticeably, we are not even able to guess which the majority class is for this dataset (it is 'acc' instead of 'unacc'). In this case, quantification works well for some classes and poorly for others. Consequently, the goodness of quantification in classification can be interpreted in different ways, according to the goal of the quantification problem (estimating the frequency of one class, calculating the majority class, deriving the Pareto ordering of classes or deriving the whole distribution). Finally, it is interesting to take a look at Figure 3 (bottom, left). We see that the classification model applied to the training dataset (which was used for building the model) does not yield perfect quantification either. In fact, we can see that the model neglects class 'vgood' while overestimating 'unacc'. This is due to the imbalance of the original dataset, where 'unacc' was highly prevalent. Typically, supervised models get biased in favour of central or majority values, because it is always preferrable in terms of expected error to bet for frequent values when there is some uncertainty. While this is good for classification metrics, we have that this *bias* is ultimately translated into the test set (or even magnified, since the model has more uncertainty on the test set). Taking this into account, we might think that getting a model which gives a perfect account of the distribution for the training set is the ideal solution, but this will generally make the model incur into overfitting, and the extrapolation to the test set will be poor —since we have a distribution shift. A possible idea to escape from this dilemma is to keep using supervised models which have been devised to have good generalisation performance as usual, and try to compensate the bias over the training set with some kind of

[2] The example is elaborated, with some fictional elements, from the cars dataset in the UCI repository (Frank and Asuncion, 2010).

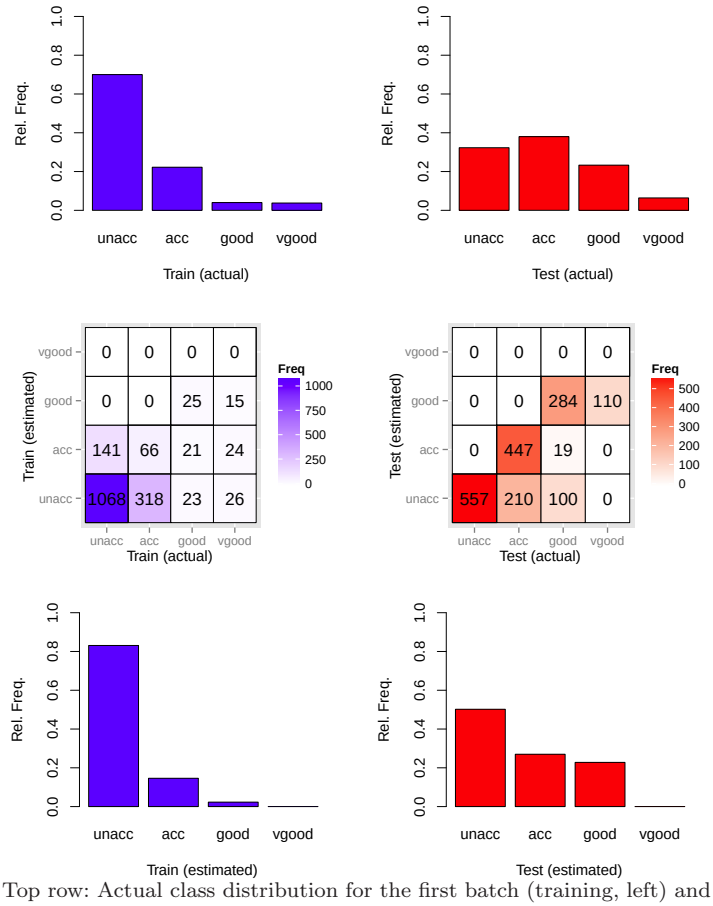

Fig. 3 Top row: Actual class distribution for the first batch (training, left) and the second batch (test, right) of Example 1. Middle row: Confusion matrices for training and test. Bottom row: Estimated class distribution using the naive quantification method *Classify & Count*.

post-hoc adjustment. This 'adjustment' is precisely what several classification quantification methods in the literature really do.

Let us now move to a regression problem.

Example 2 A quantification problem in regression
Consider a maternity ward that has collected data about baby weight at birth (dependent variable) for risk pregnancies, jointly with several features about the mother and her current and previous pregnancies (input variables). With these (training) data, a regression model has been trained in order to predict baby weight.[3] In order to better plan the resources needed and the number

[3] The example is elaborated, with some fictional elements, from the lowbwt dataset in the UCI repository (Frank and Asuncion, 2010), originally from (Hosmer and Lemeshow, 2000).

of expected complications, the hospital wants to estimate the distribution of weight births for the following month, according to a new group of pregnant women (test data) that the maternity ward is monitoring for future deliveries. This group has been reallocated from a different hospital (and borough), so we expect an important distribution shift but no important concept drift (see Figure 1), since the factors for delivery weights used in the model are assumed to have similar effects to any woman. So, the training and test distributions are different, mostly on their location, as seen in the first row of Figure 4. The second row of the figure shows how the model (a regression tree) behaves for the reference (train) and new (test) group of pregnant women. As we can see, this is not an excellent model; although there is some correlation between the actual values y and the estimated values $\hat{y}$, there seems to be some overfitting and underfitting, depending on the region (this is possible in regression trees). Note that the distribution of the estimated values differs from the distribution of the actual values for the training set (Figure 4, third row, left). The model compacts the predictions by paying less attention to those values which are at less dense regions or farther from the central part of the distribution. Also, the shape is now surprisingly bimodal, showing how distribution can be deeply modified just for the training set. In this case, because the original training distribution is relatively even (nearly symmetric), all this scarcely affects the location statistics for the estimated values, such as the mean or the median (2.50 and 2.25 versus 2.50 and 2.45 respectively).

When we compare the actual distribution for the test set (Figure 4, top right), with the naive approach by *Regress & Splice* (RS) (Figure 4, third row right), we see that the measures of location (mean and median) and spread (sd), as well as the distributions are different. Location is biased, variance is much lower and the shape is much more compact than the original. As we will see in section 4, we can improve the estimated location and spread by using several methods. Figure 4 (bottom left and right) shows better mean and standard deviation given by the method akM (on the left using a smoothed distribution as in section 4.4, and on the right an ad-hoc smoothing over the mean assuming a normal distribution). In particular, the method akM uses an *adjustment* (correcting the bias), a *segmentation* (addressing the distribution estimation locally, by segments) and a *spreading* mechanism. *Adjustment* (also mentioned in the classification quantification above) is justified because the bias is expected to be replicated (or magnified) in the test set (as happens for the RS method in this case, at least for the median, which is 0.20 kg lower for the training set but almost 0.40 kg lower for the test set). *Segmentation* is justified to better account for all the regions for which we have data in the training set, more independently of their density. *Spreading*, as mentioned above, is justified because the distribution is too compact. The reason is easy to understand and is similar to the classification case. The use of the *MSE* (mean squared error) as the metric for evaluating regression models makes that those predictions which highly deviate from the true value are strongly penalised. Consequently, for the most uncertain cases (which may come from the least populated regions) the model tends to output values closer to the

global mean. As a result, many (if not all) regression methods compact the predictions. We can see that, in this example, it is clearly the case (Figure 4, second and third rows). We will see *adjustment*, *segmentation* and *spreading* in subsequent sections. For the moment, we just want to highlight the importance of a good distribution estimation. For instance, the one on the right of the bottom row of Fig. 4 is obviously more accurate for questions such as "how many births have a weight between 2,5 and 3,5 kg", which can be answered with a value (45%), which is closer to the actual value (48%). Note that the other distribution estimation based on the method *akM* gives a value of 21%, while the estimation given by the RS method is 93%.

The previous two examples show the nature of quantification when derived by aggregating a predictive model and brings out the similarities and differences between quantification for classification and regression. These examples have also introduced some of the phenomena (bias and compactation) of the trivial aggregation methods *RS*. Both problems are shared by classification and regression, whenever the dataset is 'uneven' (in terms of class imbalance or in terms of irregular densities). Models tend to ignore peripheral (minority) cases, and this may lead to bias and compactation in the training data, which will also be present (and possibly worsened) on the test data. These previous insights are useful to better understand some previous techniques that have been developed for classification quantification, as we see below.

2.3 Previous methods for quantification

As mentioned in the introduction, the estimation of the class distribution for an unlabelled dataset has been addressed under different perspectives and applications. We mentioned some works on two-phase sampling, which referred to the problem as class prevalence estimation (Neyman, 1938; Tenenbein, 1970; Alonzo et al, 2003), where the goal and procedures were different from the setting we consider here. In some of these works, there was no distribution shift, but the need of estimating the class distribution of a population from a small sample (see Figure 1, top row). Also the estimation of the class distribution was not made by aggregating the predictions of a base classifier, but using a 'measurement device' (Neyman, 1938). In fact, this presentation of the problem is so frequent that it might have been solved in one way or another in the past, in different areas, especially from a Bayesian point of view (see, e.g., Chan and Ng 2006).

In the cases where we have a quantification problem as given by definition 1, with a distribution shift and an underlying supervised model constructed from the training set whose predictions can be aggregated for the test set, we have the setting first explored by (Forman, 2005, 2006, 2008). He developed different quantification methods (for classification) using the class predictions given by a crisp classifier or a ranker (a soft classifier outputting scores, probabilities or other estimations of the reliability of each class). The simplest one is the

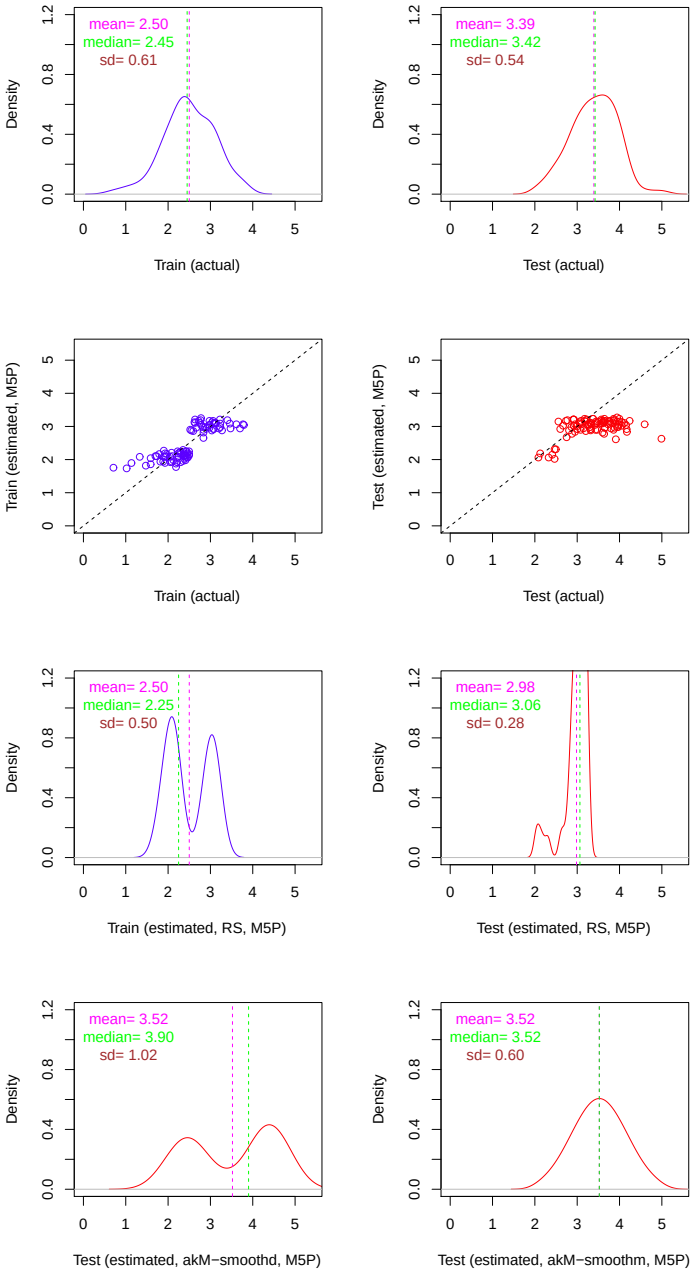

Fig. 4 Estimations using a *regression tree* for Example 2. First row: actual distributions of the dependent value (y) for the training (left) and test (right) datasets. Second row: a plot of the regression model showing the correspondence between the actual values and the estimated values for the training and test datasets. Third row: estimated class distribution using the naive quantification method *Regress & Splice* (RS) for the training and test datasets. Fourth row: estimated class distribution using a more sophisticated method akM for the test dataset: Left: using smoothing on the distribution. Right: using smoothing on the mean.

classify & count (CC) method (see Eq. 4). This method gives poor results since it underestimates the minority classes, as we have seen in Example 1. For this reason Forman introduced several other quantification methods by properly adjusting the threshold and, in some cases, also scaling the result. The *adjusted count* (AC) method is an improvement of the CC method that 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 $\frac{\sum_{i \in Test} I(\hat{y}_i = \oplus)}{|Test|}$. Forman proposed estimating the true positive rate tpr and false positive rate fpr by cross-validation on the training set. Since this scaling can give negative results or results above 1, the last step was to clip $\widehat{pos}$ to the range [0..1].

Forman also defined a collection of methods based on selecting a classifier threshold over a soft classifier which, unlike the AC method, are determined from the relationship between tpr and fpr in order to provide better quantification estimates. For instance, 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 the $T50$ method (which selects the threshold where $tpr = 50\%$) are some of the methods in this group. The *Median Sweep* (MS) method is a different approach that tests all the thresholds in the test set, estimates the number of positives in each one and returns a mean or median of these estimations. Finally, Forman proposed the *Mixture Model* (MM) (Forman, 2005) which calculated the distributions for the positive examples and negative examples separately and then joined them in a mixture. One of the conclusions of Forman's works is that the best results were obtained with MS.

There have been a variety of methods using the probability estimations of a soft classifier (Sánchez et al, 2008; Bella et al, 2010; González-Castro et al, 2012). The first method in (González-Castro et al, 2012), HDx, is not an aggregative quantification method, because it does not use a base classifier, but just works on the distribution of the input variables x. It compares likelihoods $p(x|y)$ (the authors use the term "class probability density functions") for a validation and the test dataset with the Hellinger distance (using binning to approximate the integral in the definition of this distance). For a range of class proportions, it chooses the one which leads to the smallest distance. Since it discretises the input space, this method has limitations because of data sparsity and computational cost, when the number of features is high. An alternative aggregative quantification method, HDy, is also introduced (as an adaptation of HDx), which discretises the conditional probabilities for y (instead of x). This makes it tractable when the number of features is high (and it also gives better results), because it constructs bins for y instead of x. Both methods rely on exploring a range of different values for the estimated class distribution, which works well for two classes.

Other methods use quantification to improve classification results or take advantage of a semi-supervised scenario (see, e.g, Xue and Weiss 2009). The

quantification methods in (Xue and Weiss, 2009) are mostly based on Forman's original technique.

In (Bella et al, 2010) a collection of new quantification methods based on using the class membership probability (given by a probabilistic classifier) is introduced. The idea is based on the simple *Probability estimation & Average* (PA) method, where the class probability estimations are just averaged. This can be seen as just the probabilistic version of CC, since it considers the estimated class probability for each example instead of crisp decisions. Given a probabilistic classifier $\hat{p}(y|x)$, the average of the estimated probabilities for the positive class is calculated as:

$$\hat{p}^{PA}_{Test}(\oplus) \triangleq \frac{\sum_{x \in Test} \hat{p}(\oplus|x)}{|Test|}$$

Logically, as in the CC method, if the proportion of positive examples in the training set is different from the proportion of positive examples in the test set, the result obtained by the PA method will not be satisfactory in general. So, as in the AC method, the idea is to use a proper scaling. The *Scaled Probability Average* (SPA) method (Bella et al, 2010) consists in applying the scaling over the test set that makes that the positive probability average for the positives in the training set, denoted by $Train_{\oplus}$, is 1 and that the positive probability average for the negatives in the training set, denoted by $Train_{\ominus}$, is 0. Therefore, this scaling transforms the estimation given by PA so that the *positive probability average for the positives* ($\hat{p}_{Train_{\oplus}}(\oplus)$) is 1 and the *positive probability average for the negatives* ($\hat{p}_{Train_{\ominus}}(\oplus)$) is 0. Formally, SPA is defined as:

$$\hat{p}^{SPA}_{Test}(\oplus) \triangleq \frac{\hat{p}^{PA}_{Test}(\oplus) - \hat{p}_{Train_{\ominus}}(\oplus)}{\hat{p}_{Train_{\oplus}}(\oplus) - \hat{p}_{Train_{\ominus}}(\oplus)}$$

The results in (Bella et al, 2010) show a significant improvement over Forman's methods.

3 Beyond classification: a comprehensive view of quantification and its evaluation metrics

The examples and the previous work seen in section 2 suggest that the quantification problem is multifaceted. Consequently, it can be studied according to several characteristics. This analysis leads to a comprehensive taxonomy and a set of evaluation metrics for each case, as we present in this section.

3.1 A taxonomy of quantification problems

We will consider two characteristics which critically determine the quantification problem. First, as already seen, quantification can be defined whenever we have a supervised dataset, be it a classification or regression dataset. In

Table 1 Taxonomy of quantification tasks.

In boldface, the problems and metrics undertaken in this paper.

	Task	Quantification output	Evaluation
$Q_{\mathcal{CI}}$	Classification	$\mathbb{I}[Y]$	Accuracy
$Q_{\mathcal{CD}}$	Classification	$p(Y)$	MSE, AE, MRE
$Q_{\mathcal{RI}}$	**Regression**	$\mathbb{I}[Y]$	MSE, **SE**
$Q_{\mathcal{RD}}$	**Regression**	$p(Y)$	**Cramér–von Mises** u

fact, other predictive tasks such as categorisation, hierarchical classification or ordinal regression can also lead to quantification problems.

A second characteristic is how much detail about the aggregated output we require. For instance, we may only be interested in the expected value of the output, or just a single indicator $\mathbb{I}$ (any summary statistic, such as measure of location or spread, or some other function of the distribution). For instance, in classification, this could be the mode, or majority class. In regression, this could be the mean, as in Eq. 3 or 6, or the median. In other cases, however, we may require a full distribution of the output value, which is a categorical distribution in classification, as in Eq. 1 or 4 and a continuous distribution in regression, as in Eq. 2 or 5. This dimension allows us to apply a variety of different methods to solve the quantification problem. The distinction between indicators and distribution is motivated by the fact that it is usually easier to estimate a good indicator than to estimate the whole distribution well, because the latter requires more information, as seen in Fig. 2, and the techniques may take this into account[4].

We will consider two options for the two characteristics above (classification and regression problems, and the quantification goal as a single indicator or a whole distribution), which will be represented by the letters $\mathcal{C}|\mathcal{R}$ for classification or regression, and $\mathcal{I}|\mathcal{D}$ for indicator or distribution. This gives four possible combinations and leads to a taxonomy of quantification tasks as shown in Table 1. This taxonomy broadens the scope of quantification, and can be useful for distinguishing research context for future reference.

For each problem in the taxonomy, different quantification *methods* can be defined depending on the underlying predictive model used to estimate the quantification output. It is not the same to solve quantification problems when we only have the estimation of the output value ($\hat{y}$) for each example as when we have posterior probability estimates provided by the model ($\hat{p}(y|x)$). Many classification techniques today are able to generate (soft) probabilistic models, and there are techniques for calibrating (Platt, 1999; Zadrozny and Elkan, 2001; Bella et al, 2009a,b, 2012) these probabilities which may have a positive effect on the quantification task. However, most regression techniques are still usually crisp and only output the estimation $\hat{y}$. It is true that some techniques can accompany each single prediction with the standard error, a reliability measure or a confidence band, but it is not clear how to incorporate

[4] In this paper the methodology for indicators and distribution is the same (except for some minor specific techniques, mostly at the end of the process), but this could be different in the view that some indicators require less information and effort than the whole distribution.

this information in the quantification problem. For instance, a conditional density estimator $\hat{p}(y|x)$ would lead to the understanding of quantification for regression as a distribution mixture.

In this paper we focus on methods for solving cases $Q_{\mathcal{RI}}$ and $Q_{\mathcal{RD}}$ using crisp models. We exclude from this paper those methods based on probability estimations because they require a soft regression model or a conditional density estimator $\hat{f}(y|x)$ (Hwang et al, 1994; Hyndman et al, 1996). These estimators are usually non-parametric (as the shape of the true distribution is not known). As a result, many are prone to overfitting for small or medium-sized datasets (Hwang et al, 1994), and suffering from a number of limitations (for instance, some approaches are restricted to only one or two input variables, such as R's `hdrcde` package, Hyndman et al 1996, and they do not handle nominal variables appropriately). This goes beyond the usual situation a data mining practitioner may face with a data mining tool, where she usually works with predictive models when training data is presented in a supervised fashion with attributes of many different types and possibly missing values. Typically, supervised (crisp) regression models are more robust (and faster) in these scenarios.

Finally, the taxonomy is also useful to clarify which evaluation metric is used for each case, as we see below.

3.2 Quantification evaluation metrics

We will start by reviewing the evaluation metrics for classification quantification. This set of metrics is shown in the last column of Table 1. Many previous works (Forman, 2005, 2006, 2008) on quantification for classification have used the absolute error (AE) for problem $Q_{\mathcal{CD}}$, sometimes referred to as mean absolute error (MAE), when aggregated over several repetitions or datasets. More precisely, the global AE for each class is the absolute difference of the proportion of elements for each class j in a test set $Test$ ($\hat{p}_{Test}(j)$) and the actual value:

$$AE_{Test}(j) \triangleq |p_{Test}(j) - \hat{p}_{Test}(j)|$$

and for all the classes we have the *macro-average* value (equal to the *micro-average* value for two classes):

$$AE_{Test} \triangleq \frac{1}{c} \sum_{j=1..c} AE_{Test}(j)$$

Any metric will provide a particular different view of the deviation from the actual value, and variants exists especially for normalising results when several repetitions and datasets are used in an experimental setting. Other metrics used for classification quantification are the (mean) relative error (MRE) (González-Castro et al, 2012), which is equal to AE divided by the true positive proportion, or the squared error (SE), known as MSE in (Bella et al, 2010). Finally, since in classification we really estimate probabilities, a natural choice

might seem to use measures to compare probabilities, such as cross entropy. Clearly, the choice of a particular metric has its pros and cons, and may differ in being symmetric or not about the errors and paying more or less attention to the minority classes.

In case we are interested in determining which the majority class is (a single indicator, $Q_{\mathcal{CI}}$), we suggest the use of a true/false metric indicating whether the majority class has been identified (similarly for other indicators such as the minority case). Note that a quantifier with a good AE (or SE) does not necessary imply that the majority class is correctly identified, especially in multi-class problems.

This overview of metrics for classification quantification suggests that we may also have many different options for regression quantification. We will first consider the problems which output the mean (or other indicator) for the output value ($Q_{\mathcal{RI}}$ in Table 1). Since this is a numerical value which needs to be compared to the actual mean, we use a typical measure for assessing the deviation with respect to a magnitude, the squared error. If we denote by $\mathbb{I}(Y_{Test})$ the true value of the indicator (e.g., the mean, the median, etc.) for the test or deployment dataset, and $\mathbb{I}(\hat{Y}_{Test})$ the estimated value for the same indicator, we define the squared error as follows:

$$SE_{Test} \triangleq \left(\mathbb{I}(Y_{Test}) - \mathbb{I}(\hat{Y}_{Test})\right)^2$$

For practical reasons, especially when the measure is used for an experimental evaluation of many repetitions and datasets (as in this paper), we may prefer to normalise the above measure to make values being less dependent of the magnitude range of the data and more commensurable among different datasets. In this paper, we will use the *Squared Error* (SE) seen above but normalised by the variance of the training set $Train$, denoted by VSE:

$$VSE_{Test} \triangleq \frac{\left(\mathbb{I}(Y_{Test}) - \mathbb{I}(\hat{Y}_{Test})\right)^2}{Var_{Train}(Y)}$$

This normalisation by the variance is useful when magnitudes are aggregated for several repetitions, datasets or techniques. In pairwise statistical comparisons, however, this normalisation has no effect.

Finally, we need to determine an appropriate evaluation metric for case $Q_{\mathcal{RD}}$ in Table 1. Since we need to compare the estimated distribution with the true one, we need metrics for comparing distributions, usually called divergences. However, many of them cannot be applied to empirical distributions, because the density function in some places equals 0. Consequently, empirical distributions are compared by using their cumulative distribution functions. A very simple statistic for comparing two empirical cumulative distributions F_V and F_W is the two-sample Kolmogorov-Smirnoff (KS) statistic, which is defined as:

$$KS \triangleq \max_t |F_V(t) - F_W(t)|$$

However, since the two-sample KS statistic is only based on the point where both distributions differ most, it disregards the shapes of the distributions. A more refined alternative is to take an average (or an integral), instead of a maximum. This is just the Cramér–von Mises statistic (two samples) (Anderson, 1962). In particular, from the L1–version (Xiao et al, 2006a,b) we just need the U value, which can be easily calculated from the empirical data $Y_V = v_1, v_2, \ldots, v_n$ and $Y_W = w_1, w_2, \ldots, w_m$ as follows:

Let us consider that Y_V and Y_W are sorted in increasing order. We define by $Y_{VW} \triangleq Y_V \cup Y_W$ and we also consider it is sorted. Let $r_1, r_2, \ldots, r_n$ be the ranks of the elements of Y_V in Y_{VW} and let $s_1, s_2, \ldots, s_m$ be the ranks of the elements of Y_W in Y_{VW}. Then:

$$U \triangleq n \sum_{i=1}^{n} (r_i - i)^2 + m \sum_{j=1}^{m} (s_j - j)^2$$

which is normalised as $u = \frac{U}{nm(n+m)}$. This value u (the u-statistic) will be referred to as the *cvmu* metric, which ranges between 0 and 2 (4/3 when n is large). It is lower the more similar the two distributions are. For instance, the *cvmu* metric of the distributions on the right plots of the third and fourth rows in Figure 4 with respect to the true distribution for the test set (top right plot) are 0.41 and 0.18, respectively.

4 Regression quantification methods

Now that the place of quantification for regression in the family of quantification tasks has been clarified, as well as its evaluation metrics, we are ready to focus on problems $Q_{\mathcal{RI}}$ and $Q_{\mathcal{RD}}$ in Table 1. As in the classification case, quantification is meaningless for those applications where the training and test distributions always match. In fact, if this were the case, the perfect solution for $Q_{\mathcal{RI}}$ in Table 1 would be just to estimate the indicator on the training set. For instance, if the indicator is the mean, the *Test to Train* (TT) method just assigns the "same mean", simply defined as:

$$\hat{\mu}^{TT}_{Test} \triangleq \mu_{Train}$$

where μ_{Train} is an instance of Eq. 3 for the training set. Clearly, this can be adapted for any indicator, such as the median. In the experiments in section 5 we will use the same acronym TT, for the corresponding "same median" version of the method. This method is just included as a baseline or reference, since we will focus on methods which account for a distribution shift.

Similarly, we can define a method called *Test to Train* (also denoted by TT), which just 'copies' the distribution from the train to the test set and will be used as baseline for the $Q_{\mathcal{RD}}$ problem. The method just uses the y values in the training as the empirical distribution for the test. Note that there is no mapping between examples (as in other methods where the predictions are modified). In fact, the sizes of the training and test are usually different (which is not a problem for the *cvmu* metric seen in the previous section).

4.1 Analysing overfitting, bias and variance

Before presenting some more elaborated techniques, we analyse the regression quantification problem better to see how to take advantage from an underlying supervised model. Typically, regression models are trained (and evaluated) to minimise their mean squared error (*MSE*). From here, we can analyse the performance of a model with the classical (see, e.g., Hastie et al 2009; Flach 2012) bias-variance decomposition of *one* example for *all* possible datasets D:

$$\begin{aligned}\mathbb{E}_{\{D\}}[(y-\hat{y})^2] &= (\mathbb{E}_{\{D\}}[\hat{y}]-y)^2 + \mathbb{E}_{\{D\}}[(\hat{y}-\mathbb{E}_{\{D\}}[\hat{y}])^2] \\ &= (\mathbb{E}_{\{D\}}[\hat{y}-y])^2 + \mathbb{E}_{\{D\}}[(\hat{y}-\mathbb{E}_{\{D\}}[\hat{y}])^2] \\ &= (Bias_{\{D\}}(\hat{y}-y))^2 + Var_{\{D\}}(\hat{y})\end{aligned}$$

where $\mathbb{E}_{\{D\}}$ denotes the expected value for all possible datasets. The above decomposition is usually a way to understanding overfitting as high variance: the *predictions* vary very significantly when we change the dataset, i.e., when we move from training to test. Underfitting is usually understood as high bias. At first sight, it may seem that a good regression model for quantification needs to have low bias. However, we cannot ignore the variance, especially because we have a data distribution shift. So, also for quantification, a good compromise between overfitting and underfitting must be found, because both are harmful for the extrapolation for new unseen areas. One thing that can be observed from the previous decomposition is the effect of outermost predictions. One can think that outermost overestimations are not harmful provided they are usually accompanied by a balanced proportion and magnitude of outermost underestimations. While this might be true for non-quadratic errors because they cancel, it is not true for *MSE*, as shown by both components. This is the reason why most regression techniques output predictions whose variance (i.e., $Var(\hat{y})$) is lower than the actual variance ($Var(y)$), where Var here refers to the variance of *all* the examples in *one* dataset. This phenomenon seriously affects the spread of the predictions, leading to more packed predictions. As a result, regression models trained to minimise the *MSE* will need to be spread out (the shape of the distribution should be widened) in order to resemble the actual distribution, as we saw in Figure 4. The method introduced in section 4.4 is precisely based on smoothing the distribution such that $Var(\hat{y}) = Var(y)$.

While the previous decomposition gives us some understanding about spread, it gives us few clues about the location of the estimated distribution. For this purpose, it is more insightful to make the same decomposition of *all* examples on *one* dataset, since it is the result for all examples what counts for quantification, as follows:

$$\mathbb{E}[(y-\hat{y})^2] = \mathbb{E}[\epsilon^2]$$

$$= (\mathbb{E}[\epsilon])^2 + \mathbb{E}[(\epsilon - \mathbb{E}[\epsilon])^2]$$
$$= (Bias(\epsilon))^2 + Var(\epsilon)$$

where $\mathbb{E}$ denotes the expected value for *all* the examples in *one* dataset, and the error is denoted by $\epsilon = y - \hat{y}$. Note that now the variance refers to the *errors*, not the predictions. Just focussing on one dataset, we cannot get information about overfitting and underfitting, but we can see other phenomena. We see that if the model gives $Bias(\epsilon) = 0$ then the mean of the errors will be zero — even if the errors may still be non-zero individually. This zero bias means that the mean will be perfectly estimated for that dataset. If the indicator function we are interested in is the mean, then we would get perfect quantification. Consequently, we want regression models such that $Bias(\epsilon) = 0$. We can see clearly that if we add a constant s to all the predictions, we get a different $Bias(\epsilon)$ but equal $Var(\epsilon)$. It is then natural to expect that many regression techniques try to set $Bias(\epsilon) = 0$ by calibrating the model in this way. In fact, some linear regression techniques ensure $Bias(\epsilon) = 0$ on the training set by definition, because the MSE is minimised[5]. Other techniques, however, may give slightly uncalibrated models for the training set, because the asymmetry of the training set forces the technique to unbalance the estimations for risk minimisation. Independently of the regression technique being used, we can always find the optimal constant s for a dataset, by just setting s equal to the bias:

$$s = Bias(\epsilon) = \mathbb{E}[\epsilon] = \mathbb{E}[y - \hat{y}] = \mathbb{E}[y] - \mathbb{E}[\hat{y}] \tag{7}$$

which just subtracts the mean of the actual values with the mean of the estimated values. It is also expectable that since the errors are usually higher on a test set, this value s may be higher for the test set than for the training set. This will be the basis of the adjustment methods below.

Finally, while a global adjustment may solve some cases, the bias can vary significantly between the central areas of the distribution and the outermost values. Typically, outermost values, as mentioned above, are usually pushed to the centre. This means that low y values will typically lead to negative bias, and high y values will typically lead to positive bias. This suggests the use of adjustment constants customised for different regions of the distribution, leading to the segmentation methods in section 4.3.

[5] As a mean-unbiased estimator minimises squared loss, a median-unbiased estimator is a different choice which minimises the absolute error.

4.2 Methods based on *Regress & Sum*

The simplest method for estimating the mean using an underlying regression model is the *Regress & Sum* (RS) method we defined in Eq. 6 but applied to the test set, which we will denote by $\hat{\mu}^{RS}_{Test}$. The RS method can handle a distribution shift, but it depends on the quality of the training data sample (it must be representative of the overall domain) as well as on the quality of the regression model. Otherwise, the estimation might even be worse than the TT method. This is similar to the problems already observed for the *Classify & Count* method in classification quantification.

In a quite similar way as the AC method (Forman, 2008) is an adjusted improvement of the CC method, or as SPA is a scaled adjustment of the naive PA method (Bella et al, 2010), we can follow the same idea in regression. In order to do this, we need to calculate the average of the *true* values for the $Train$ set, μ_{Train}. We also need to calculate the average of the estimated values, i.e., $\hat{\mu}^{RS}_{Train}$ (an instance of Eq. 6 for the training set). Now, we can derive the *error bias* (*Bias* above, in what follows denoted by B) for the training set as follows:

$$B^{RS}_{Train} \triangleq \mu_{Train} - \hat{\mu}^{RS}_{Train}$$

With this value B we can adjust any method, following Eq. 7. For instance, the *Adjusted Regress and Sum* method (aRS) is just:

$$\hat{\mu}^{aRS}_{Test} \triangleq \hat{\mu}^{RS}_{Test} + \alpha \cdot B^{RS}_{Train} \tag{8}$$

where α is a parameter that makes the adjustment more or less intense, motivated by errors being expected to increase for the test set, as discussed in the previous section. This value can be estimated from the use of a regression technique on similar problems, or can be set to a fixed constant independently of technique and problem, as we do in this paper. Note that when $\alpha = 0$ this is equivalent to the RS method.

Figure 4.2 shows a simple example which illustrates how the methods described in this section work. We use a simple regression problem with only one attribute. We split the data into two sets of the same size, $Train$ and $Test$, but different distribution. On the left side of the figure we can see the $Train$ dataset (points as blue circles). The dotted red line represents the model built from the training data using a linear regression method. The solid black horizontal line represents the average of the actual values y for the instances in the $Train$ set (0.071). The dashed red line shows the average of the linear regression model (0.031). The difference between these values indicates that the regression model is not calibrated with respect to the training instances. This slight difference will be used by the aRS technique. On the right side of Figure 4.2 we include the result of the application of the RS method over the $Test$ set and the results of using the aRS (with $\alpha = 1$) method with the same test dataset. We can see the different data distribution between $Train$ (mean 0.07) and $Test$ (mean 0.44). In this figure, the TT and RS methods are

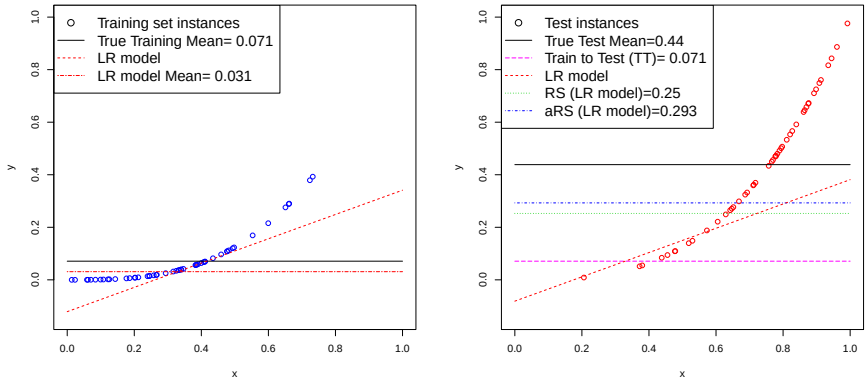

Fig. 5 An artificial example $y = x^3$ showing how several methods in section 4.2 work. Training and test are sampled from 100 examples with a triangular distributions over x in [0.1] with a 50/50 proportion. Left: Train data (mean = 0.07) and the regression model. Right: Test data (mean = 0.44). The RS method and the aRS method (with $\alpha = 1$) are shown.

shown. The solid line expresses the value of the mean of the test dataset, i.e., the actual average value. The RS method using the linear regression returns a value (0.25), shown as a green dotted horizontal line, which is slightly below the actual one (0.44). However, it is much better than the TT method (0.07). Finally, the blue dashed-and-dotted horizontal line shows the result of using the aRS (with $\alpha = 1$) method with the same test dataset. Given that the linear regression is not calibrated with respect to the training data, there is a correction that modifies the estimated value for this technique to 0.293.

The previous methods RS and aRS have been presented to address the $Q_{\mathcal{RI}}$ quantification task in Table 1 where the indicator is the mean. The idea can be easily extended for the median and other indicators. Also, the RS and the aRS methods can be extended for the $Q_{\mathcal{RD}}$ case, where the whole distribution needs to be estimated, and spread and shape are also important. For this extension, we only need to apply Eq. 5, which was called *Regress & Splice*. This intentionally leads to the same acronym RS, since basically both methods are identical. In fact, Figure 4 (third row, right) shows the results for RS for the test set in Example 2. While this actually gives a distribution and not a single indicator, we see that the distribution has low dispersion and most of the data clusters around the mean.

4.3 Methods based on segmentation

The previous methods are simple adaptations of some of the most common methods in classification quantification. One of the problems when a distribution shift takes place is that some ranges of values of the dependent value which appear in the test set are rare in the training set. Consequently, the

underlying model is not well trained for these ranges, and the quantification approach by RS methods just worsens this issue, especially if there are outliers either in the training or the test set. A second problem, which appears in the $Q_{\mathcal{RD}}$ case (Table 1), is that the estimated distributions have low dispersion and a highly peaked shape, and applying a correction in the same direction is not going to have any effect on this issue. A third, related problem is that some ranges may have more bias than others. In other words, it is a strong assumption to think that the bias is uniform all along the range of values. In fact, it may even be positive in some regions and negative in others, precisely because models usually compact their predictions.

One solution for these three problems is to make a local adjustment to the aggregation. In other words, instead of applying a global correction obtained from the whole $Train$ set (as the aRS method does) we propose to adjust the estimated value of each instance in the $Test$ set by using only a suitable subset of the training instances (a bin).[6] Therefore, we propose a segmentation of the output values in the $Train$ set into several groups. From each group we derive a true average value that we compare to the average estimated value of that group using the model, in order to determine different and local values for the bias. This local difference (bias) is used to adjust the values in the group when predicting the values for the $Test$ set.

More precisely, given the set of output values y on the training set denoted by Y, we will just apply a segmentation method d (we will consider several, as we will discuss later on) that sorts the elements of Y in ascending order, creates a sequence of k consecutive bins $\{Y_1, Y_2, \ldots, Y_k\}$ and defines a sequence of $k-1$ limits or thresholds $T = \{t_1, t_2, \ldots, t_{k-1}\}$ such that the threshold t_j is calculated averaging the maximum value of the bin Y_j and the minimum value of the bin Y_{j+1}. Moreover, note that segmentation is applied to the actual values y and not to the estimated values $\hat{y}$ by the model. Then, the values for each bin are averaged giving a sequence of bin prototypes $y_1^m, y_2^m, \ldots, y_k^m$. Next, for each bin j, an estimated bin prototype $\hat{y}_j^m$ is calculated by averaging the estimated values $\hat{y}$ for the training examples in that bin. From here, and now on the test set, quantification is performed by replacing each prediction $\hat{y}$ for the test set by the estimated prototype of the bin where it belongs, i.e., if $t_j \leqslant \hat{y} < t_{j+1}$, the example belongs to bin j and its estimated prototype is $\hat{y}_j^m$ (we consider that $t_0 = -\infty$ and $t_k = \infty$ in order to cover all possible values). Finally, from these modified predictions a common RS method can be applied. This method can be used with any kind of regression model and keeps quantification general and simple.

[6] The idea of segmenting the set of outputs is not new and has led to some classifier calibration techniques, such as binning (Zadrozny and Elkan, 2002; Bella et al, 2009b). Calibration techniques are somewhat related to quantification techniques. In fact, RS would be optimal if the predictive model were perfectly calibrated —for the *test* set. This is a key point because calibration is always understood relative to a distribution or dataset. Given the quantification problems with distribution shift we are considering here, it is the test set distribution what we want to infer, so calibrating for the training set may be useless.

Figure 6 illustrates this process. On the training set (left) a segmentation method has generated three bins (with the same number of examples in each bin). The average values of each bin are, $y_1^m = 0.32$, $y_2^m = 0.63$ and $y_3^m = 0.87$. The threshold between the first and the second bin is $t_1 = 0.46$, and the threshold between the second and the third bin is $t_2 = 0.79$. The estimated average values for each bin are $\hat{y}_1^m = 0.33$, $\hat{y}_2^m = 0.52$ and $\hat{y}_3^m = 0.85$. Therefore, when the value predicted by the learning algorithm on the test set (right) for an instance is less than or equal to 0.46 the method assigns 0.33 to it. If the value predicted by the learning algorithm of an instance is between 0.46 and 0.79 the method assigns 0.52 to it. Finally, if the value is greater than 0.79 the method assigns 0.85. Clearly, segmentation does not provide enough granularity for instance regression, but here we are interested in quantification.

Train set

Ins.	y	y^m_j	t_j	$\hat{y}$	$\hat{y}^m_j$	B_j
3	0.23	0.32		0.22	0.33	- 0.01
7	0.30			0.41		
1	0.44		← 0.46	0.37		
4	0.48	0.63		0.35	0.52	0.11
8	0.67			0.52		
9	0.75		← 0.79	0.68		
2	0.84	0.87		0.89	0.85	0.02
5	0.84			0.74		
6	0.93			0.92		

Test set

Ins.	$\hat{y}$	$\hat{y}^m_j$	$\hat{y}^a$ (α=5)
1	0.33	0.33	0.28
2	0.91	0.85	0.95
3	0.66	0.52	1.07
4	0.45	0.33	0.28
5	0.77	0.52	1.07

Fig. 6 Example of the predictions of a regression model for a test set (right) using the segmentation on the training set (left). On the left we show how the bins are constructed and how the bias B_j is estimated for each bin using the difference $y_j^m - \hat{y}_j^m$. On the right, each example is assigned to a bin j according to its prediction $\hat{y}$ and the final adjusted prediction $\hat{y}^a$ is given by Eq. 9 using $\alpha = 5$.

The example is also useful to see that segmentation has some effects on the distribution of the new output values. Segmentation makes prediction less sensitive to outliers and, depending on how the bins are chosen, the values might be more robust than a single prediction. In fact, it may have important and non-monotonic effects when the distribution is multimodal. Also, segmentation has the effect of reducing variability, especially if k is small.

From this process, we can use the modified (segmented) predictions as the new numerical values for each example, which can be used for both *RS* (Eq. 5 and 6) as seen in section 2.1. However, the *interesting* thing about the segmentation method comes when we combine it with an adjustment[7], as justified by the third problem we have mentioned above: we cannot assume

[7] Note that this adjustment is performed with information from the training data exclusively. An alternative possibility would be to use a validation dataset, but this would reduce the available training data.

that the bias B_j is the same for all the regions. So, instead of calculating a global adjustment as in Eq. 8, we calculate the bias for each bin in the training set, and we use it for the adjustment for that bin. In other words, we adjust each bin independently of the rest. This has much more impact on how predictions are modified, since some bins may have very poor estimated averages. As can be seen in Figure 6 we have calculated the mean $\hat{y}_j^m$ of the predicted values $\hat{y}$ on the training set of each bin j, and B_j is calculated for each bin as $(y_j^m - \hat{y}_j^m)$. From here, the adjusted value is calculated for each instance in the test set as follows:

$$\hat{y}^a = \hat{y}_j^m + \alpha \cdot B_j \tag{9}$$

In Figure 6 we consider $\alpha = 5$, and we obtain $0.33 + 5 \cdot (-0.01) = 0.28$ for example $i = 1$, $0.85 + 5 \cdot (0.11) = 0.95$ for example $i = 2$ and so on. Note that many regression models will tend to issue predictions leaned towards the global average (because they are unsure on some cases and also because of the *MSE* penalisation, discussed in section 4.1). This will lead to different values of B_j for each bin, trying to set the values outwards.

In order to apply the previous procedure, we need some segmentation methods. Two methods will derive the segments using discretisation techniques and the third one will use a clustering method. The three of them will keep the numerical character of the output value (we are not discretising the problem). Bakar et al. (Bakar et al, 2009) present an exhaustive taxonomy for data discretisation techniques. We will first explore two of the simplest and best-known unsupervised methods: *Equal Frequency intervals* (EF) and *Equal Width intervals* (EW) (Dougherty et al, 1995). Basically, these methods consist in sorting the values and splitting them in k bins. The number of bins is a parameter that has to be supplied by the user. EW puts the same number of examples in each bin whereas EF creates bins with the same length: $(max\_value - min\_value)/k$. A third method ($kMs$) is not based on discretisation techniques and will segment the output values by using *k-means*, which is one of the most popular clustering algorithms. The k-means algorithm is applied on the outputs, as in the other methods, so it finally creates a partition, which may be different to the other two cases.

All these three segmentation methods require a value for k. There are several formulae to automatically set the number of bins. For example, Sturges (Sturges, 1926) sets $k = log_2(n + 1)$, where n is the number of instances. Yang (Yang, 2003) proposes a new discretisation method called *Proportional Discretisation* (PD) that combines the EF and EW methods. In PD, the number of examples in each bin (frequency) is equal to the number of bins, and the frequency multiplied by the number of bins is equal to the number of examples. Therefore, this method is equivalent to using the EF method with $k = \sqrt{n}$. We have studied these two alternatives for establishing k. Since the results are quite similar for both options, in this paper we will only include the results for $k = log_2(n + 1)$. For each segmentation method, we will also consider an unadjusted and an adjusted version. This leads to 6 combinations.

Table 2 shows these 6 different quantification methods for regression problems based on segmentation.

Table 2 Quantification methods for regression based on segmentation.

	Segmentation method	Number of bins	Representative value of each bin
EFl	Equal Frequency	$k = log_2(n+1)$	mean
EWl	Equal Width	$k = log_2(n+1)$	mean
kMl	kMeans	$k = log_2(n+1)$	mean
aEFl	Equal Frequency	$k = log_2(n+1)$	mean adjusted with B
aEWl	Equal Width	$k = log_2(n+1)$	mean adjusted with B
akMl	kMeans	$k = log_2(n+1)$	mean adjusted with B

It is important to notice that all these methods are extremely easy to apply, and some of them can just be computed directly by using data mining tools (these tools typically include many discretisation and clustering methods, such as kMeans, which can be used for the segmentation).

4.4 Spreading the distribution

In section 4.1 we discussed that regression methods usually get very packed predictions, where it is rarely the case that $Var(\hat{y}) = Var(y)$. We also saw this in Example 2. While this problem does not affect the mean estimation significantly, it can be important for some other indicators ($Q_{\mathcal{RI}}$) such as the median or other quantiles. Also, this is especially problematic for the $Q_{\mathcal{RD}}$ quantification problem in Table 1.

The segmentation process using adjustment may spread the distribution slightly but it can be insufficient in general. In order to get an appropriate degree of spreading, we apply a kernel smoothing[8] of the estimated values $\hat{y}$. We use the *ksmooth* method in the R project software (R Team et al, 2012) with default parameters. If $Var(\hat{y}) < Var(y)$ (as usual), we adjust the bandwidth until we get $Var(\hat{y}) = Var(y)$. If $Var(\hat{y}) \geq Var(y)$ we just use the smoothing with the default bandwidth. After that, we get new values for the predictions using as many quantiles as needed.

All the methods above, including TT, RS and aRS, will be altered by this smoothing procedure as a postprocess.

4.5 Regression quantification using classification quantification

Now we include some methods based on discretising the original regression problem and applying classification quantification techniques to the modified

[8] An alternative, more lightweight approach could be to introduce a normal jitter to each prediction $\hat{y}$. While this may have a similar effect, it has random effects that may be important for small datasets. The smoothing approach presented here always leads to the same result, since it has no random components.

problem, and transforming it back to regression again. This indirect way of addressing regression quantification is just included as a reference.

Note that here we no longer work with regression models but with classification models. This means that the methods in this subsection are not applicable when we want to take advantage of an existing (possibly validated) regression model. Also note that it is different to segment the distribution given by the regression model (as we have done above) than to completely discretising the problem from scratch as we discuss here. Specifically, we will show how to apply three classification quantification methods to a regression quantification problem which has been converted into a classification problem from the beginning, by discretising the training data. As all the methods we will use (AC, $T50$ and MS methods, Forman 2008) were defined for binary problems, and the discretisation of a regression output generally generates more than two classes, we will need to apply the *one-vs-all* approach for the multi-class case as Forman suggested. Although this *one-vs-all* approach mangles all the ordinal information[9], we explore whether this can be alternative to native regression quantification methods, and which of the previous classification quantification methods could be best suited in this case.

A simple way to discretise the problem is to use the segmentation on the training set obtained by applying one of the segmentation techniques we introduced in section 4.3. The idea is to assign a different class label l_j to each one of the c bins defined by the segmentation process. Next we have a classification training dataset from which we train a classification model. For the classification model we record the mean for each class on the training set, denoted as μ_j. Using the classification model we then use any of the classification quantification methods in the literature. Each method will output a probability for each class l_j as $\hat{p}_{T_Y}(l_j)$ for the test set T. The discrete predictions are then converted back into continuous predictions by setting μ_j if the predicted class is l_j. Consequently, the estimated mean for the test set will be $\sum_{j=1}^{c} \mu_j \cdot \hat{p}_{T_Y}(l_j)$. Other indicators can be obtained from pointwise (or binwise) quantification using distribution smoothing techniques. In the following experimental section we will just evaluate the mean for these classification quantification methods for regression quantification.

5 Experiments

In this section, we present the experimental evaluation of the methods for regression quantification introduced in the previous section: those based on *Regress & Sum* or *Splice*, namely RS and aRS, and the six methods based on data segmentation. We will examine the results for the quantification problems $Q_{\mathcal{RI}}$ (mean and median) and $Q_{\mathcal{RD}}$ in Table 1 and we will use their

[9] Some alternatives could be figure out here, such as the use of *one-vs-previous* or *one-vs-adjacent* schemes. This is left as a possibility for future work.

corresponding metrics (VSE and Cramér–von Mises u metric, $cvmu$). Finally, we will also include some results of the approach to regression quantification using classification quantification.

5.1 Experimental setting

The experimental setting is based on the common case where we train a regression model on a training dataset, and we want to obtain an indicator ($Q_{\mathcal{RI}}$) of the output value for a different test dataset or the whole distribution ($Q_{\mathcal{RD}}$). As justified in previous sections, we are especially interested in cases where the distribution of the output value varies between training and test. Additionally, it is important to consider a scenario where the distribution of the dependent value varies significantly, because it also affects the quality of the model, see (Weiss, 2004; Weiss and Provost, 2001; Raeder et al, 2012; Chawla et al, 2004). In order to ensure that this distribution shift takes place in the experiments, for each dataset, we construct a test set selecting the 50% of the instances by using a discrete triangular distribution over the example indices, after sorting the examples according to their output value. More precisely, the instances are sorted increasing by its output attribute leading to sorted indices i, and a probability of $i/(n+1)$ is assigned to each instance, with $i = 1 \dots n$, where n is the number of instances of the dataset. Note that since the distribution is applied over the indices, we do not have to care about outliers and the range of the data, and it even works in cases where the distribution is packed. Using this distribution, 50% examples are sampled without substitution for the test set. The instances that have not been selected for the test set form the training set. With this, we get a training set which misrepresents the high values, and a test set which focusses precisely on these high values. Figure 7 shows this.

We selected 35 datasets from the UCI repository[10] (Frank and Asuncion, 2010), Tunedit[11] and mldata[12]. In Table 3 we show the name of the datasets, the number of attributes, the number of instances of the datasets, and the mean and the standard deviation of the output value. We also include the means for the training and test datasets.

The whole battery of datasets can be downloaded from *http://alturl.com/5n8ad*. In order to evaluate several regression techniques, we employed five data mining techniques from *WEKA* (Witten and Frank, 2005): *Linear Regression*, *M5P* (a regression tree), *SMOreg* (a support vector machine), *IBk* (a nearest neighbour with k=10) and *Gaussian* (a Gaussian process regression). We used the default parameters in *WEKA* for the five algorithms. We repeated each configuration 100 times for each dataset. This makes a total of $35 \times 5 \times 100 = 17{,}500$ experiments for each quantification method.

[10] *http://archive.ics.uci.edu/ml/*
[11] *http://tunedit.org/repo*
[12] *http://mldata.org/repository/data/*

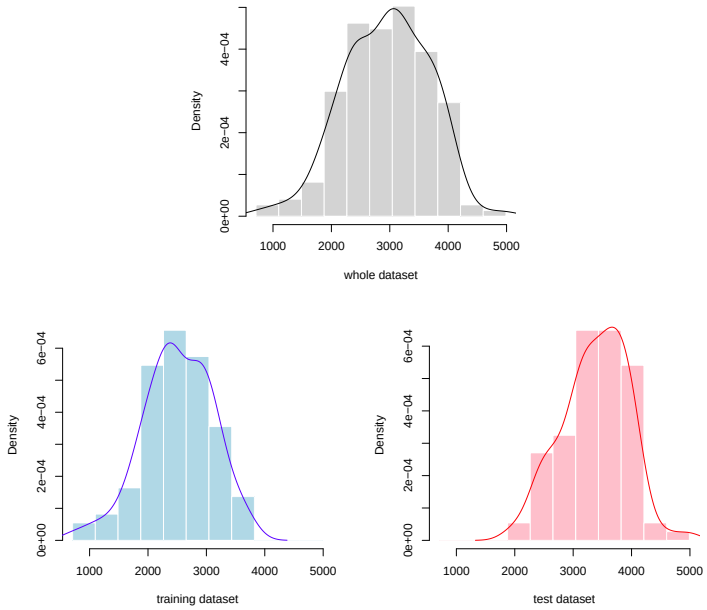

Fig. 7 Sampling process to simulate a data distribution shift for the experiments. Top: the distribution for dataset lowbwt. Bottom: the distributions of training and test datasets after sampling using a triangular distribution over the sorted indices.

Before starting with the experiments, we studied which value of α leads to the best adjustment. For this reason, we performed an experimental evaluation varying the value of α from 0 (no correction) to 7. In these experiments we employed 10 datasets of Table 3 with 10 repetitions each.

These results showed that a value of 5 obtains the optimal performance in almost all cases. Consequently, we will use this value of α in the following experiments.

In the experiments that follow we work with several indicators. For the mean indicator we just calculated the mean as usual, and for the distribution we just aggregated the results. For every application of the median (bins, global, etc.) we used the Hodges-Lehmann estimator (Hodges and Lehmann, 1963) to give more robust estimations of the median.

5.2 General overview

We start with a set of highly aggregated results showing the behaviour of all the regression quantification methods and all the indicators we will use (mean, median and *cvmu* metric). The goal of this first overview is to get a first impression of how methods work and, most especially, to see the general effect of the spreading, adjustment and segmentation.

Table 3 Datasets used in the experiments. The columns represent the name, number of attributes, number of examples, the number of different values, the mean, median and standard deviation for the whole dataset and the mean for the Training and test datasets.

	Dataset	#Att.	Size	Distinct	Mean	Median	SD	MeanTrain	MeanTest
1	gascons	5	27	26	207.03	225.8	43.8	186.68	228.95
2	bolts	8	40	40	33.93	19.26	27.41	20.89	46.98
3	vineyard	4	52	19	18.09	19.5	4.39	16	20.17
4	elusage	3	55	52	43.28	38.62	24.01	31.66	55.33
5	pollution	16	60	60	940.36	943.68	62.21	908.30	972.42
6	mbagrade	3	61	57	3.29	3.32	0.33	3.13	3.46
7	auto93	23	93	81	19.51	17.7	9.66	15.05	24.06
8	baskball	5	96	95	0.42	0.43	0.11	0.36	0.48
9	cloud	7	108	94	1.23	0.91	1.08	0.73	1.73
10	fruitfly	5	125	47	23.46	20	15.88	15.85	31.2
11	echoMonths	10	130	53	22.18	23.5	15.86	13.74	30.62
12	veteran	8	137	101	121.63	80	157.82	55.99	188.23
13	fishcatch	8	158	101	398.7	272.5	359.09	212.3	585.09
14	autoPrice	16	159	145	11445.73	9233	5877.86	8570.18	14357.68
15	servo	5	167	51	1.39	0.73	1.56	0.72	2.07
16	lowbwt	10	189	133	2944.66	2977	729.02	2554.1	3339.37
17	pharynx	12	195	177	558.73	445	418.72	346.47	773.18
18	pwLinear	11	200	189	-0.34	-0.56	4.47	-2.76	2.08
19	cpu	8	209	104	99.33	45	154.76	47.48	151.68
20	bodyfat	15	252	176	19.15	19.2	8.37	14.54	23.76
21	breastTumor	10	286	23	24.66	25	10.36	19.11	30.2
22	cholesterol	14	303	152	246.69	241	51.78	220.12	273.44
23	autoMpg	8	398	129	23.51	23	7.82	19.22	27.81
24	pbc	19	418	399	1917.78	1730	1104.67	1305.7	2529.87
25	housing	14	506	229	22.53	21.2	9.2	17.79	27.28
26	sensory	12	576	11	15.07	15	0.82	14.63	15.52
27	strike	7	625	358	302.3	129	560.66	102.16	503.08
28	quakes(stations)	5	1000	102	33.42	27	21.9	22.52	44.32
29	quakes(mag)	5	1000	22	4.62	4.6	0.4	4.4	4.84
30	concrete(str.)	9	1030	845	35.82	34.45	16.71	26.56	45.08
31	concrete(cem.)	9	1030	278	281.17	272.9	104.51	223.25	339.08
32	quake(ritcher)	4	2178	12	5.98	5.9	0.19	5.88	6.07
33	quake(focal)	4	2178	312	74.36	39	116.47	30.46	118.26
34	parkins.(motor)	17	5875	1080	21.3	20.87	8.13	16.68	25.91
35	parkins.(total)	17	5875	1129	29.02	27.58	10.7	23.03	35.01

Table 4 Aggregated results (5 regression techniques × 35 datasets × 100 iterations = 17,500 values) for each indicator (mean, median and *cvmu* metric) and quantification technique. We show methods TT (same distribution for $Test$ as $Train$), RS (*Regress & Sum* or *Regress & Splice* depending on the case), aRS (the adjusted version of RS), followed by the six methods based on binning, three without adjustment (EW, EF, kM) and three with adjustment (aEW, aEF, akM). There are rows which incorporate smoothing as a spreading method and rows which do not, as indicated.

	TT	RS	aRS	EW	aEW	EF	aEF	kM	akM
Mean w/o smoothing (Avg. VSE)	1.24	0.66	0.69	0.76	0.49	0.75	0.39	0.75	0.45
Mean with smoothing (Avg. VSE)	1.26	0.67	0.70	0.77	0.50	0.77	0.40	0.77	0.46
Median w/o smoothing (Avg. VSE)	1.24	0.58	0.77	0.68	0.54	0.67	0.52	0.68	0.54
Median with smoothing (Avg. VSE)	1.26	0.59	0.79	0.71	0.49	0.71	0.45	0.71	0.48
Distrib. w/o smoothing (Avg. *cvmu*)	0.85	0.60	0.58	0.52	0.29	0.52	0.23	0.51	0.24
Distrib. with smoothing (Avg. *cvmu*)	0.43	0.22	0.21	0.28	0.19	0.27	0.15	0.28	0.17

Table 4 shows the results for the three indicators using the metrics as introduced in section 3.2. From these general and highly aggregated results we have a first impression of the quantification methods. The direct method RS works much better than inheriting the distribution from the training set (TT), since for this experimental setting we have created an important distribution shift. This was expected and quite clear from the results of the two first columns.

From the table we can also see that the adjustment over RS (i.e., aRS) is not effective (it is either comparable or worse than RS), so the idea of the adjustment, per se, does not seem to improve the quantification results. Then we can focus on the three methods based on binning (EW, EF, kM) and we see no improvement over RS, except a slight improvement for the distribution case without smoothing. It is the conjunction of binning and adjustment that leads to good results, as shown with the three methods based on binning and adjustment (aEW, aEF, akM).

Finally, from this big picture, we are interested in the effect of smoothing. As expected, it has almost no effect on the mean, since a Gaussian smoothing is usually quite conservative for this indicator. For the median, smoothing is effective for the three methods with binning and adjustment while for the rest of methods it has not a significant effect. Finally, for the whole distribution, smoothing is extremely effective for all methods. From this general observation, we will just concentrate in the results with smoothing in the rest of this section.

Given this first look at the results, we now start a series of more detailed analyses according to several issues.

5.3 Analysis for the mean indicator

Let us now focus on the experimental results for $Q_{\mathcal{RI}}$ using the mean. Table 5 shows the comparison of the VSE metric for all the datasets in terms of pairwise tests, using RS as reference. Note that the use of VSE or SE is irrelevant here, since the denominator (the variance) is always the same for the two things we compare. For each dataset and pairwise comparison for the 100 repetitions we use the Wilcoxon's signed-rank test. For the overall results at the bottom, and in order to see whether the difference between more than two methods is statistically significant, we calculated the Friedman test with the Iman and Davenport modification. The tests are applied to the average ranks for the 35 datasets. If there were differences between these methods, we calculated the Nemenyi post-hoc test to compare all of the methods with each other (with a probability of 99.5%) as recommended in (Demsar, 2006).

From the results in Table 4 (row with "Mean with smoothing") and Table 5, we see that the aRS method, while significantly better in terms of pairwise statistical comparison is slightly worse than RS on average. So it is not clear whether the global adjustment is really improving the results. However, the local adjustment is really useful for the segmentation methods. The three segmentation methods have a similar behaviour, but from Table 4 (row with "Mean with smoothing") and Table 5 the best method is clearly aEF (pairwise significant and with the smallest average). It also outperforms the RS method in 29 datasets and only loses in 5 datasets (1 tie).

If we focus on dataset size there seems to be no relation between the size of the datasets and the results, since no clear trend is found which can distinguish the datasets from the top to those on the bottom. Note that they are sorted by dataset size (see Table 3). We could also consider whether the

Table 5 Comparison for the *mean* indicator for all the quantification methods against method *RS*. The values (W/T/L) represent the number of wins, ties and losses respectively (in boldface we indicate whether the difference is significant with a probability of 99.5% using the Wilcoxons signed-rank test). The five regression techniques are aggregated, so we make 5×100 comparisons in each cell of the table. All methods incorporate smoothing as a spreading method. The rows W, T and L show all the comparisons ($35 \times 5 \times 100 = 17{,}500$). The final row (R) shows the average rank for the Nemenyi test and whether the differences are significant: in boldface if the method is better than RS, underlined when the method is worse than RS, and normal face when there is no statistical difference.

	aRS v RS	EW v RS	aEW v RS	EF v RS	aEF v RS	kM v RS	akM v RS
1	254/25/221	141/0/**359**	**316**/0/184	135/1/**364**	**299**/0/201	149/1/**350**	**302**/0/198
2	**286**/46/168	34/0/**466**	249/0/251	51/0/**449**	**266**/0/234	38/0/**462**	**296**/0/204
3	74/111/**315**	79/0/**421**	286/0/214	33/0/**467**	262/0/238	62/0/**438**	263/0/237
4	**396**/21/83	26/0/**474**	**372**/0/128	49/0/**451**	**354**/0/146	32/0/**468**	**381**/0/119
5	189/31/**280**	3/0/**497**	**352**/0/148	6/0/**494**	**348**/0/152	5/0/**495**	**353**/0/147
6	115/63/**322**	33/16/**451**	**422**/0/78	43/18/**439**	**401**/0/99	33/16/**451**	**401**/0/99
7	**344**/1/155	35/73/**392**	**364**/0/136	103/73/**324**	**353**/0/147	43/73/**384**	**368**/0/132
8	164/61/**275**	0/0/**500**	**483**/0/17	2/0/**498**	**448**/0/52	1/0/**499**	**479**/0/21
9	**266**/45/189	3/0/**497**	**361**/0/139	13/0/**487**	**302**/0/198	3/0/**497**	**376**/0/124
10	**286**/7/207	**285**/64/151	209/0/**291**	**311**/64/125	231/0/**269**	**293**/64/143	198/0/**302**
11	268/0/232	195/0/**305**	**294**/0/206	222/0/**278**	**328**/0/172	198/0/**302**	**344**/0/156
12	**312**/4/184	3/0/**497**	**368**/0/132	29/0/**471**	**470**/0/30	14/0/**486**	**472**/0/28
13	237/0/**263**	87/0/**413**	**268**/0/232	86/0/**414**	238/0/**262**	70/0/**430**	**288**/0/212
14	**306**/10/184	2/0/**498**	**427**/0/73	25/0/**475**	**383**/0/117	2/0/**498**	**433**/0/67
15	**379**/25/96	56/0/**444**	214/0/**286**	40/0/**460**	**369**/0/131	73/0/**427**	241/0/**259**
16	236/24/**240**	32/0/**468**	**346**/0/154	174/0/**326**	**417**/0/83	142/0/**358**	248/0/**252**
17	221/0/279	90/6/**404**	**282**/0/218	233/6/**261**	**415**/0/85	107/6/**387**	**330**/0/170
18	**327**/11/162	0/0/**500**	**356**/0/144	0/0/**500**	**430**/0/70	0/0/**500**	**360**/0/140
19	**273**/6/221	9/0/**491**	**288**/0/212	15/0/**485**	164/0/**336**	15/0/**485**	**276**/0/224
20	**373**/1/126	0/0/**500**	248/0/252	0/0/**500**	111/0/**389**	0/0/**500**	192/0/**308**
21	**435**/0/65	21/0/**479**	**433**/0/67	60/0/**440**	**493**/0/7	32/0/**468**	**449**/0/51
22	117/0/**383**	53/1/**446**	**394**/0/106	49/1/**450**	**450**/0/50	55/1/**444**	**419**/0/81
23	**363**/1/136	0/0/**500**	**485**/0/15	1/0/**499**	**380**/0/120	0/0/**500**	**470**/0/30
24	220/0/280	3/0/**497**	**455**/0/45	5/0/**495**	**494**/0/6	2/0/**498**	**489**/0/11
25	**481**/1/18	0/0/**500**	**489**/0/11	0/0/**500**	**439**/0/61	0/0/**500**	**491**/0/9
26	**294**/45/161	0/0/**500**	**497**/0/3	0/0/**500**	**496**/0/4	1/0/**499**	**390**/0/110
27	**402**/2/96	0/0/**500**	**470**/0/30	5/0/**495**	**496**/0/4	1/0/**499**	**497**/0/3
28	**351**/0/149	0/0/**500**	**443**/0/57	0/0/**500**	**393**/0/107	0/0/**500**	**425**/0/75
29	**320**/6/174	0/0/**500**	110/0/**390**	0/0/**500**	65/0/**435**	0/0/**500**	110/0/**390**
30	182/12/**306**	0/0/**500**	**326**/0/174	0/0/**500**	**377**/0/123	0/0/**500**	**346**/0/154
31	**444**/1/55	0/0/**500**	240/0/**260**	0/0/**500**	**297**/0/203	0/0/**500**	**279**/0/221
32	**282**/5/213	51/93/**356**	99/0/**401**	106/93/**301**	**347**/0/153	91/93/**316**	212/0/**288**
33	**353**/0/147	0/0/**500**	5/0/**495**	11/0/**489**	**476**/0/24	9/0/**491**	**403**/0/97
34	**456**/0/44	0/0/**500**	**489**/0/11	2/0/**498**	**498**/0/2	0/0/**500**	**498**/0/2
35	**401**/4/95	0/0/**500**	**500**/0/0	0/0/**500**	**500**/0/0	0/0/**500**	**500**/0/0
W	**10407**	1241	**11940**	1809	**12790**	1471	**12579**
T	569	253	0	256	0	254	0
L	6524	**16006**	5560	**15435**	4710	**15775**	4921
R	**3.88**	6.42	**3.41**	6.02	**2.93**	6.19	**3.13**

number of distinct values (the 'distinct' column in Table 3) has any effect. The datasets where the number of different values is smallest (≤ 25) are datasets 3, 21, 26, 29 and 32. Actually, these datasets could be just considered ordinal regression problems instead of proper regression problems. Again, we see no special trend or behaviour for these five datasets, so the methods behave quite homogeneously in this regard.

Finally, in Table 6 we disaggregate the results of Table 4 (row with "Mean with smoothing") by the underlying regression technique. While the values are different in magnitude, the general picture is preserved in all of them, with aEF being the best method.

Table 6 Aggregated results (35 datasets × 100 iterations = 3,500 values) for the mean indicator as in Table 4 (row 'Mean with smoothing') disaggregating by the five base regression techniques.

	TT	RS	aRS	EW	aEW	EF	aEF	kM	akM
LR	1.26	0.89	0.88	0.70	0.58	0.71	0.40	0.70	0.51
M5P	1.26	0.48	0.53	0.63	0.48	0.64	0.30	0.62	0.43
SMO	1.26	0.67	0.55	0.75	0.36	0.75	0.37	0.75	0.37
Gaussian	1.26	0.58	0.90	0.77	0.52	0.75	0.40	0.76	0.47
IBk	1.26	0.75	0.65	1.02	0.55	0.98	0.52	1.01	0.54
Avg.	1.26	0.67	0.70	0.77	0.50	0.77	0.40	0.77	0.46

5.4 Analysis for the median indicator

We continue with $Q_{\mathcal{RI}}$ but now using the median indicator, which is a more robust statistic. Table 7 shows the comparison of the VSE metric for all the datasets in terms of pairwise tests, using RS as reference. Note that the use of VSE or SE is also irrelevant here, since the denominator (the variance) is always the same for the two things we compare. We use the same statistical tests as we used for the mean.

From the results in Table 4 (row with "Median with smoothing") and Table 7 we see that the aRS method is now significantly worse than RS on average (as happened for the mean) and also in terms of pairwise statistical comparison. So the global adjustment is not improving the results. However, the local adjustment is really useful for the segmentation methods, as happened for the mean. The three segmentation methods have a similar behaviour, but from Table 4 (row with "Median with smoothing") and Table 7 the best method is clearly aEF (pairwise significant and with a smaller average). It also outperforms the RS method in 20 datasets and only loses in 9 datasets (6 ties). The difference is not so strong as it was for the mean, possibly because the Median is a more robust indicator than the mean.

If we focus on dataset size or number of repeated values, there is no special phenomenon, so it seems that the methods behave quite homogeneously. Finally, in Table 8 we disaggregate the results of Table 4 (row with "Median with smoothing") by the underlying regression technique. While the values are different in magnitude, the general picture is preserved in all of them, with aEF being the best method, as happened with the mean, except for SMO.

5.5 Analysis for the whole distribution

Now we change to the experimental results for the $Q_{\mathcal{RD}}$ quantification task, where we are interested in how well the whole distribution is estimated. Table 9 shows the comparison in terms of the *cvmu* metric for all the datasets by pairwise comparison, using RS as reference. We use the same statistical tests as we used for the mean and median cases. We can see that the best methods for $Q_{\mathcal{RI}}$ are also the best methods for $Q_{\mathcal{RD}}$.

From the results in Table 4 (row with "Distrib. with smoothing") and Table 9 we see that binning with local adjustment is really useful. The three

Table 7 Comparison for the *median* indicator for all the quantification methods against method *RS*. The configuration and interpretation of rows and statistical tests are as in Table 5.

	aRS v RS	EW v RS	aEW v RS	EF v RS	aEF v RS	kM v RS	akM v RS
1	257/0/243	84/0/**416**	**282**/0/218	71/1/**428**	252/0/248	98/1/**401**	261/0/239
2	207/0/**293**	48/0/**452**	219/0/**281**	71/0/**429**	217/0/**283**	54/0/**446**	**295**/0/205
3	273/1/226	111/0/**389**	302/0/198	63/0/**437**	246/0/**254**	100/0/**400**	285/0/215
4	**281**/0/219	64/0/**436**	**346**/0/154	94/0/**406**	**335**/0/165	56/0/**444**	**329**/0/171
5	258/0/242	6/0/**494**	**335**/0/165	5/0/**495**	**343**/0/157	3/0/**497**	**332**/0/168
6	124/0/**376**	38/22/**440**	**409**/0/91	34/20/**446**	**381**/0/119	37/19/**444**	**400**/0/100
7	216/0/**284**	38/73/**389**	**311**/0/189	93/73/**334**	300/0/200	41/73/**386**	**304**/0/196
8	232/0/**268**	0/0/**500**	**463**/0/37	3/0/**497**	**438**/0/62	0/0/**500**	**455**/0/45
9	90/0/**410**	15/0/**485**	260/0/240	20/0/**480**	225/0/**275**	13/0/**487**	**300**/0/200
10	106/3/**391**	**269**/84/147	227/0/**273**	**270**/80/150	206/0/**294**	**268**/79/153	205/0/**295**
11	215/0/**285**	85/0/**415**	246/0/**254**	71/1/**428**	282/0/218	64/0/**436**	275/0/225
12	81/0/**419**	7/0/**493**	**342**/0/158	47/0/**453**	**441**/0/59	19/0/**481**	**433**/0/67
13	71/0/**429**	105/1/**394**	235/0/265	104/0/**396**	194/0/**306**	86/0/**414**	222/0/**278**
14	119/0/**381**	15/0/**485**	**337**/0/163	38/0/**462**	280/0/220	14/0/**486**	**325**/0/175
15	60/0/**440**	57/0/**443**	179/0/**321**	47/0/**453**	170/0/**330**	62/0/**438**	191/0/**309**
16	139/0/**361**	1/0/**499**	233/0/**267**	5/0/**495**	**352**/0/148	5/0/**495**	174/0/**326**
17	94/0/**406**	65/6/**429**	**274**/0/226	197/6/**297**	**403**/0/97	101/6/**393**	**308**/0/192
18	**317**/0/183	0/0/**500**	**345**/0/155	0/0/**500**	**404**/0/96	0/0/**500**	**361**/0/139
19	133/3/**364**	79/0/**421**	234/0/**266**	55/0/**445**	109/0/**391**	60/0/**440**	219/0/**281**
20	**255**/1/244	0/0/**500**	232/0/268	0/0/**500**	115/0/**385**	0/0/**500**	199/0/**301**
21	**485**/0/15	35/0/**465**	**406**/0/94	100/0/**400**	**490**/0/10	49/0/**451**	**420**/0/80
22	86/1/**413**	48/6/**446**	**355**/0/145	47/5/**448**	**452**/0/48	49/5/**446**	**409**/0/91
23	20/0/**480**	0/0/**500**	**452**/0/48	0/0/**500**	**422**/0/78	0/0/**500**	**448**/0/52
24	86/0/**414**	2/0/**498**	**385**/0/115	3/0/**497**	**473**/0/27	2/0/**498**	**444**/0/56
25	252/0/248	0/0/**500**	**466**/0/34	0/0/**500**	**420**/0/80	0/0/**500**	**461**/0/39
26	360/0/140	0/0/**500**	**498**/0/2	0/0/**500**	**457**/0/43	1/0/**499**	**366**/0/134
27	71/0/**429**	0/0/**500**	**458**/0/42	4/0/**496**	**458**/0/42	4/0/**496**	**490**/0/10
28	81/7/**412**	0/0/**500**	**395**/0/105	0/0/**500**	264/0/236	0/0/**500**	**369**/0/131
29	**450**/1/49	0/0/**500**	27/0/**473**	0/0/**500**	29/0/**471**	0/0/**500**	32/0/**468**
30	**397**/0/103	0/0/**500**	227/0/273	0/0/**500**	219/0/281	0/0/**500**	234/0/266
31	191/0/**309**	0/0/**500**	**310**/0/190	0/0/**500**	**367**/0/133	0/0/**500**	**344**/0/156
32	99/0/**401**	57/97/**346**	99/0/**401**	110/94/**296**	**394**/0/106	102/94/**304**	146/0/**354**
33	100/0/**400**	0/0/**500**	3/0/**497**	12/0/**488**	**427**/0/73	10/0/**490**	**358**/0/142
34	100/0/**400**	0/0/**500**	**413**/0/87	41/0/**459**	**466**/0/34	2/0/**498**	**446**/0/54
35	100/0/**400**	0/0/**500**	**475**/0/25	0/0/**500**	**489**/0/11	0/0/**500**	**467**/0/33
W	6406	1229	**10780**	1605	**11520**	1300	**11307**
T	17	289	0	280	0	277	0
L	**11077**	**15982**	6720	**15615**	5980	**15923**	6193
R	4.89	6	3.55	5.59	**3.29**	5.78	**3.35**

Table 8 Aggregated results (35 datasets × 100 iterations = 3,500 values) for the median indicator as in Table 4 (row 'Median with smoothing') disaggregating by the five base regression techniques.

	TT	RS	aRS	EW	aEW	EF	aEF	kM	akM
LR	1.26	0.75	0.90	0.65	0.57	0.66	0.46	0.65	0.53
M5P	1.26	0.43	0.71	0.57	0.47	0.59	0.35	0.57	0.45
SMO	1.26	0.56	0.55	0.69	0.37	0.69	0.42	0.68	0.41
Gaussian	1.26	0.52	0.99	0.71	0.50	0.70	0.47	0.70	0.49
IBk	1.26	0.67	0.79	0.95	0.54	0.92	0.55	0.93	0.54
AVG.	1.26	0.59	0.79	0.71	0.49	0.71	0.45	0.71	0.48

segmentation methods have a similar behaviour, but from Table 4 (row with "Distrib. with smoothing") and Table 7 the best method is aEF in terms of average but akM in terms of pairwise comparison. Model aEF also outperforms the RS method in 26 datasets and only loses in 9 datasets (0 ties), and aKM wins in 27 and loses in 6 (2 ties). If we focus on dataset size or number of repeated values, again there seems to be no relation, as happened with the mean and the median.

Table 9 Comparison for the *cvmu* indicator for all the quantification methods against method *RS*. The configuration and interpretation of rows and statistical tests are as in Table 5.

	aRS v RS	EW v RS	aEW v RS	EF v RS	aEF v RS	kM v RS	akM v RS
1	188/172/140	142/12/**346**	**315**/12/173	129/18/**353**	**299**/12/189	151/16/**333**	**301**/5/194
2	**175**/178/147	185/0/**315**	**326**/1/173	79/9/**412**	**303**/0/197	186/0/**314**	**415**/0/85
3	29/183/**288**	76/23/**401**	185/0/**315**	39/24/**437**	152/0/**348**	58/18/**424**	201/2/**297**
4	**309**/126/65	38/2/**460**	**302**/0/198	51/4/**445**	**313**/1/186	33/0/**467**	**337**/0/163
5	120/181/**199**	2/0/**498**	273/0/227	7/0/**493**	**278**/0/222	3/0/**497**	259/0/241
6	29/201/**270**	30/20/**450**	**432**/0/68	43/21/**436**	**391**/1/108	30/22/**448**	**400**/0/100
7	**341**/5/154	40/73/**387**	**310**/1/189	103/73/**324**	**289**/0/211	44/73/**383**	**314**/0/186
8	117/200/**183**	0/0/**500**	**461**/0/39	1/0/**499**	**459**/0/41	1/0/**499**	**457**/0/43
9	149/190/161	66/0/**434**	**280**/0/220	53/0/**447**	174/0/**326**	62/0/**438**	**301**/0/199
10	138/202/160	**282**/74/144	**325**/0/175	**303**/70/127	**339**/0/161	**289**/70/141	**303**/0/197
11	268/1/231	193/0/**307**	**330**/0/170	222/2/**276**	**353**/0/147	199/3/**298**	**359**/0/141
12	**235**/177/88	3/0/**497**	**386**/1/113	27/0/**473**	**435**/0/65	13/0/**487**	**439**/0/61
13	223/0/**277**	129/0/**371**	**297**/0/203	67/0/**433**	191/0/**309**	108/0/**392**	267/0/233
14	**208**/140/152	103/0/**397**	**394**/0/106	25/0/**475**	**307**/0/193	103/0/**397**	**392**/0/108
15	153/172/175	231/0/**269**	109/0/**391**	203/0/**297**	182/0/**318**	224/0/**276**	171/0/**329**
16	133/192/**175**	55/0/**445**	**328**/0/172	256/0/244	238/0/**262**	198/0/**302**	249/0/**251**
17	207/29/264	89/6/**405**	**277**/0/223	230/6/**264**	**408**/0/92	94/6/**400**	**300**/0/200
18	156/196/148	0/0/**500**	**332**/0/168	0/0/**500**	**274**/0/226	0/0/**500**	**317**/0/183
19	**245**/112/143	**376**/0/124	**403**/0/97	**370**/0/130	**365**/0/135	**387**/0/113	**439**/0/61
20	**284**/158/58	2/0/**498**	172/0/**328**	0/0/**500**	108/0/**392**	0/0/**500**	148/0/**352**
21	**434**/8/58	22/0/**478**	**490**/0/10	57/4/**439**	**499**/0/1	31/1/**468**	**475**/0/25
22	118/0/**382**	51/1/**448**	**450**/0/50	44/1/**455**	**481**/0/19	52/1/**447**	**489**/0/11
23	**357**/10/133	0/0/**500**	**452**/0/48	9/0/**491**	189/0/**311**	1/0/**499**	**429**/0/71
24	220/0/280	3/0/**497**	**460**/0/40	5/0/**495**	**496**/0/4	3/0/**497**	**497**/0/3
25	**385**/98/17	0/0/**500**	**456**/0/44	0/0/**500**	**251**/0/249	0/0/**500**	**452**/0/48
26	**155**/270/75	0/0/**500**	**496**/0/4	0/0/**500**	**498**/0/2	0/1/**499**	**437**/0/63
27	**316**/168/16	0/0/**500**	**322**/0/178	5/0/**495**	**381**/0/119	1/0/**499**	**418**/0/82
28	**272**/106/122	0/0/**500**	**424**/0/76	0/0/**500**	**345**/0/155	0/0/**500**	**418**/0/82
29	**301**/197/2	0/0/**500**	93/0/**407**	0/0/**500**	79/0/**421**	0/0/**500**	95/0/**405**
30	144/78/**278**	0/0/**500**	**413**/0/87	0/0/**500**	**365**/0/135	0/0/**500**	**407**/0/93
31	**331**/128/41	0/0/**500**	**324**/0/176	0/0/**500**	236/0/**264**	0/0/**500**	**328**/0/172
32	**188**/195/117	29/223/**248**	99/0/**401**	77/238/**185**	**402**/0/98	56/236/**208**	219/0/**281**
33	**294**/102/104	0/0/**500**	6/0/**494**	11/0/**489**	**388**/0/112	11/0/**489**	**326**/0/174
34	**394**/64/42	0/0/**500**	**500**/0/0	2/0/**498**	**500**/0/0	0/0/**500**	**500**/0/0
35	**349**/79/72	0/0/**500**	**500**/0/0	0/0/**500**	**500**/0/0	0/0/**500**	**500**/0/0
W	**7965**	2147	**11722**	2418	**11468**	2338	**12359**
T	4318	434	15	470	14	447	7
L	5217	**14919**	5763	**14612**	6018	**14715**	5134
R	**3.96**	6.23	**3.38**	5.81	**3.48**	5.99	**3.1**

Finally, in Table 10 we disaggregate the results of Table 4 (row with "Distrib. with smoothing") by the underlying regression technique. While the values are different in magnitude, the general picture is preserved in all of them, with aEF being the best method, except for SMO, where it is akM.

Table 10 Aggregated results (35 datasets × 100 iterations = 3,500 values) for the cvmu indicator as in Table 4 (row 'Distrib. with smoothing') disaggregating by the five base regression techniques.

	TT	RS	aRS	EW	aEW	EF	aEF	kM	akM
LR	0.43	0.22	0.22	0.26	0.22	0.26	0.16	0.25	0.19
M5P	0.43	0.19	0.20	0.23	0.20	0.24	0.14	0.23	0.17
SMO	0.43	0.22	0.17	0.28	0.15	0.27	0.16	0.28	0.14
Gaussian	0.43	0.21	0.24	0.28	0.20	0.27	0.15	0.27	0.17
IBk	0.43	0.27	0.22	0.35	0.21	0.33	0.16	0.35	0.17
AVG.	0.43	0.22	0.21	0.28	0.19	0.27	0.15	0.28	0.17

5.6 Analysis for the methods based on classification

We will now investigate the results for the methods based on classifiers, as seen in section 4.5. Here we will only focus on the mean, given the characteristics of these methods. Table 11 shows the comparison in terms of the VSE metric for all the datasets by pairwise comparison, using RS as reference. We apply the same statistical tests as we used for the previous cases. We use four underlying classifiers: J48, Logistic Regression, IBk ($k = 10$) and Naïve Bayes, all of them using Weka with default parameters. We use the process of discretising the problem and then a classification quantification method (Forman's Adjusted Count, T50 o Median Sweep), followed by a recovering of the magnitudes from the classes. Since we have three discretisation methods (EW, EF, kM), there are 9 methods. All them are compared against RS.

Table 11 Comparison for the *mean* indicator using the VSE metric for all the quantification methods based on the conversion of the problem into a classification against the genuine regression quantification method RS. There are four classification techniques and five regression techniques for RS, so we make $4 \times 5 \times 100$ comparisons in each cell of the table. The rows W and L show all the comparisons ($35 \times 4 \times 5 \times 100 = 70{,}000$). There are no ties in this table (because we are comparing very different approaches). Consequently, we only show W/L (wins and losses). The configuration of statistical tests is as in Table 5.

	ACEW-RS	ACEF-RS	ACkM-RS	T50EW-RS	T50EF-RS	T50kM-RS	MSEW-RS	MSEF-RS	MSkM-RS
1	**1283**/717	**1288**/712	**1297**/703	690/**1310**	761/**1239**	595/**1405**	859/**1141**	**1191**/809	859/**1141**
2	**1023**/977	505/**1495**	823/**1177**	933/**1067**	380/**1620**	745/**1255**	505/**1495**	645/**1355**	708/**1292**
3	318/**1682**	677/**1323**	564/**1436**	396/**1604**	211/**1789**	203/**1797**	**1407**/593	**1117**/883	614/**1386**
4	**1499**/501	895/**1105**	**1192**/808	**1146**/854	729/**1271**	849/**1151**	544/**1456**	**1538**/462	964/**1036**
5	474/**1526**	441/**1559**	504/**1496**	335/**1665**	139/**1861**	322/**1678**	774/**1226**	**1344**/656	489/**1511**
6	916/**1084**	763/**1237**	**1012**/988	656/**1344**	678/**1322**	826/**1174**	904/**1096**	734/**1266**	876/**1124**
7	**1360**/640	502/**1498**	1002/998	1053/947	311/**1689**	635/**1365**	457/**1543**	**1529**/471	653/**1347**
8	**1148**/852	1010/990	**1211**/789	714/**1286**	730/**1270**	618/**1382**	865/**1135**	**1728**/272	802/**1198**
9	692/**1308**	279/**1721**	637/**1363**	556/**1444**	147/**1853**	434/**1566**	211/**1789**	**1194**/806	428/**1572**
10	**1856**/144	**1074**/926	**1573**/427	**1299**/701	749/**1251**	**1461**/539	341/**1659**	238/**1762**	**1389**/611
11	**1637**/363	882/**1118**	**1295**/705	**1149**/851	569/**1431**	690/**1310**	793/**1207**	**1205**/795	658/**1342**
12	**1874**/126	822/**1178**	**1804**/196	**1716**/284	786/**1214**	**1446**/554	59/**1941**	**1277**/723	**1283**/717
13	980/**1020**	256/**1744**	585/**1415**	648/**1352**	148/**1852**	373/**1627**	458/**1542**	751/**1249**	429/**1571**
14	**1197**/803	153/**1847**	883/**1117**	930/**1070**	37/**1963**	487/**1513**	97/**1903**	**1637**/363	563/**1437**
15	**1551**/449	334/**1666**	**1478**/522	**1396**/604	151/**1849**	981/**1019**	26/**1974**	**1450**/550	649/**1351**
16	651/**1349**	1141/859	660/**1340**	415/**1585**	516/**1484**	197/**1803**	**1054**/946	**1782**/218	366/**1634**
17	**1888**/112	999/1001	**1480**/520	**1667**/333	788/**1212**	**1428**/572	105/**1895**	1364/636	**1342**/658
18	775/**1225**	214/**1786**	596/**1404**	399/**1601**	25/**1975**	196/**1804**	489/**1511**	**1714**/286	253/**1747**
19	973/**1027**	159/**1841**	615/**1385**	**1057**/943	5/**1995**	493/**1507**	26/**1974**	939/1061	473/**1527**
20	339/**1661**	110/**1890**	228/**1772**	73/**1927**	14/**1986**	62/**1938**	520/**1480**	742/**1258**	145/**1855**
21	**1557**/443	**1724**/276	**1297**/703	817/**1183**	680/**1320**	**1030**/970	535/**1465**	249/**1751**	**1025**/975
22	**1408**/592	878/1122	**1176**/824	**1179**/821	902/1098	**1448**/552	275/**1725**	180/**1820**	**1439**/561
23	**1268**/732	266/**1734**	874/**1126**	470/**1530**	24/**1976**	207/**1793**	493/**1507**	**1477**/523	324/**1676**
24	**1994**/6	**1141**/859	**1859**/141	**1458**/542	809/**1191**	**1202**/798	381/**1619**	779/**1221**	**1164**/836
25	**1602**/398	546/**1454**	**1445**/555	**1379**/621	72/**1928**	487/**1513**	283/**1717**	**1840**/160	521/**1479**
26	979/**1021**	**1984**/16	1105/895	858/**1142**	**1122**/878	643/**1357**	636/**1364**	229/**1771**	654/**1346**
27	**1727**/273	679/**1321**	**1860**/140	**1654**/346	787/**1213**	**1222**/778	0/**2000**	**1710**/290	930/**1070**
28	**1498**/502	491/**1509**	**1813**/187	**1578**/422	1211/789	**1464**/536	1/**1999**	**1948**/52	**1059**/941
29	**1492**/508	599/**1401**	**1476**/524	938/**1062**	471/**1529**	974/**1026**	19/**1981**	**1190**/810	686/**1314**
30	**1836**/164	674/**1326**	**1350**/650	1103/897	615/**1385**	217/**1783**	261/**1739**	870/**1130**	322/**1678**
31	**1502**/498	355/**1645**	634/**1366**	940/**1060**	280/**1720**	77/**1923**	94/**1906**	791/**1209**	291/**1709**
32	**1950**/50	**1988**/12	**1975**/25	1010/990	667/**1333**	**1777**/223	370/**1630**	562/**1438**	**1840**/160
33	559/**1441**	**1427**/573	**1997**/3	466/**1534**	987/1013	**1738**/262	64/**1936**	284/**1716**	**1655**/345
34	**1996**/4	**1170**/830	**1682**/318	821/**1179**	279/**1721**	968/**1032**	307/**1693**	192/**1808**	973/1027
35	**1997**/3	**1195**/805	**1687**/313	**1158**/842	59/**1941**	816/**1184**	**1153**/847	59/**1941**	819/**1181**
W	**45799**	27621	**41669**	33057	16839	27311	15366	**36479**	27645
L	24201	**42379**	28331	**36943**	**53161**	**42689**	**54634**	33521	**42355**
R	**3.64**	5.75	**4.23**	5.16	7.19	6.02	7.34	**4.81**	5.99

From the results in Table 11 we have to say that this indirect approach is not as ill-conceived as it may seem. The results are relatively good for some of the AC methods and MS with EF. This is surprising, especially because the T50 and MS methods do not work well. One possible explanation may originate from the fact that converting a regression problem into a classification problem

loses the ordinal information and some methods may be more sensitive than others to this fact.

In the end, the comparison between methods using four classification techniques and methods using five classification is not easy to interpret (and possibly not meaningful) because it depends on the quality of the five underlying regression techniques versus the four underlying classification techniques, and not properly on the quantification method. Nonetheless, we just include it here to see whether this approach of discretising regression problems could work.

5.7 Discussion and recommendations

As a summary of the results, we can state that the methods based on segmentation and local adjustment clearly surpass the performance of the baseline method (TT) and the simple RS method. The statistical significance tests corroborate this statement. We have also seen that some of the ideas presented in this paper work well on isolation (such as smoothing) and some others only work well together (segmentation with adjustment). The results are quite homogeneous in terms of types of datasets.

So, what is our recommendation when a regression quantification problem appears? On the one hand, if we already have a regression model, then it is reasonable to use that model for quantification using a method such as aEF. As for Table 4, this seems a good bet for almost every indicator. On the other hand, if we do not have a regression model, we may decide to find a technique which works well for regression. Here, we can take a look at Tables 6, 8 and 10, and see that M5P will lead to the best results for mean, M5P and SMO are good for the median and SMO seems better for the whole distribution. Nonetheless, this cannot be generalised for families of techniques, since there might be other regression trees, linear regression, Gaussian, kNN or SVM approaches with better results, or the same techniques by just changing a few parameters. Also, when there is no regression model, it is an option to dicretise the problem and see what happens with classification quantification, especially because the best method for regression quantification using classification seems to be AC, which is easy to implement.

6 Conclusions

Quantification can be seen as a group of closely related tasks where the goal is to determine some global indicator or the whole distribution from a set of (individual) unlabelled examples. Aggregative quantification is the view of quantification as a problem which is solved by aggregating the predictions of an underlying predictive model: a classification model if the output value is categorical, or a regression model if the output value is numerical. Where

many proposals have been recently proposed for classification quantification, to our knowledge this is the first paper addressing the problem of regression quantification, as shaped here.

Quantification is a common problem in many data mining applications:

- Multidimensional hierarchical data. Data warehouses are frequently the source of minable views. A predictive model learnt at a fine-grained level may be required to be applied to a different distribution (region, period or category) or at a different level of aggregation. Quantification techniques are crucial for this problem.
- Overall estimation. Problems are frequently presented in the form of a batch, where we need to anticipate (e.g., assign resources) an assessment of some indicators or the whole distribution for a batch of customers, patients, complaints, errors, diagnoses, etc., before making any individual decision.
- Distribution shift detection. Clearly a quantifier can be used to determine when (and of course how much) a distribution is shifting. This may be used to trigger some revision or re-training of the models, instead of triggering this by the degradation of the models (which can be originated by other factors, such as obsolescence, but not necessarily a distribution shift).
- Calibration and cost applications. An interesting application would be to use quantification methods to calibrate supervised models or to use them in cost applications, only incipiently explored by Forman (Forman, 2008). However, the methods introduced in this paper modify all the single predictions, so the model resulting from adjustment, segmentation and spreading can still be used as a single-instance supervised model (this is not the case, for instance, with methods such as T50 or MS in classification).

As discussed throughout the paper, the problem becomes really meaningful when we have an important distribution shift between the training data and the deployment data. This distribution shift entails a change in the categorical distribution (class frequencies) in classification problems. In regression, the distribution shift may involve a change in location, dispersion and shape of the data distribution. This is in contrast to the classical problem of estimating the distribution from a small sample, even if there is no distribution shift.

All the above issues have triggered a comprehensive analysis of quantification as a group of data mining tasks on its own, which is consolidated by the new taxonomy we have introduced in this paper. From here, we can see how much similar the quantification tasks are, what metrics are required for each of them and the desired detail of the quantification result, as a single indicator or a whole distribution. Given this taxonomy we have focussed on the two tasks which, to our knowledge, have not received proper attention in the literature (in the terms set here). These families of tasks are regression quantification obtaining the whole distribution and the task focussed on a particular indicator. One of the principles we have followed when looking for solutions for these problems is that the solutions should be general to any underlying regression technique.

We have seen that the direct adaptation of ideas for classification quantification to regression does not work. Moreover, the methods based on segmentation introduced here, especially when each bin is used to do local adjustments, have shown a very significant improvement. We have seen good results for both the mean and the median estimation and the whole distribution estimation ($Q_{\mathcal{RI}}$ and $Q_{\mathcal{RD}}$ tasks).

There are also some limitations of the approach presented in this paper. Aggregative regression quantification still relies on good regression models to get good quantification. While it is possible to get good quantification from some poor models, there is still much to be done in order to understand which regression models (and techniques) are more suitable for quantification. While some quantification methods excel on average, they can perform worse than the trivial RS method for some datasets and techniques. Detecting when this is the case, and using method combinations with this information could lead to improved results. Also, the results for the whole distribution case $Q_{\mathcal{RD}}$ can be improved in terms of shape, since asymmetries and multimodalities may not be properly identified in many cases.

In the end, this is just a first attempt for regression quantification (as considered here) and many things can be improved. For instance, we have set the same value for the parameter α for all techniques and methods in order to ease the comparison, but we may choose a particular α for each problem, regression technique and quantification method, or link it to some parameters of the model on the training set, such as correlation or the variance. Also, new smoothing methods could be devised as well by reusing the information from the segmentation. Segmentation could also be done in other ways, by the use of other clustering methods, sliding windows or kernels, with possibly overlapping bins.

As a side possibility, we might wonder what learning parameters and metrics should be used for training supervised models on purpose for a quantification problem. For instance, we have used regression techniques with default parameters usually aiming at minimising *MSE*, but we could also use other techniques with the purpose of being median-unbiased estimators by the use of the absolute error instead. The approaches using classification may also be considered. Some of the analysis of existing performance metrics (Ferri et al, 2009; Hernández-Orallo et al, 2012) could be overhauled having quantification in mind. The use of different loss functions and metrics could also lead to extensions of the taxonomy, by accommodating ordinal regression, hierarchical classification, etc.

As mentioned above, quantification is a very common problem, and there are also many possible scenarios and applications for the regression case. In fact, the interaction of regression quantification with common hierarchical data organisations deserves to be explored. In fact, quantification could be applied in an iterative way. For instance, we may need to calculate the expected total benefits for a city, then aggregate for regions, then for countries, etc. This stepwise aggregation may give better results than aggregating the predictions from the lowest level to the highest one. This, of course, suggests the use of

(hierarchical) clustering for both the input and the output values. In fact, many different segmentation methods could be used instead of the EF, EW and kM methods. All this could also be explored as future work.

All in all, regression quantification may have been approached in many different (and ad-hoc) ways in the literature and specific applications, but there has not been a systematic and generalised account of this data mining task on its own. Given the number of alternatives and the ideas that could be used for different task variants of the regression quantification problem, we think this paper is a resolute step in the recognition of this family of data mining tasks and the everyday application of better suited solutions.

Acknowledgements We would like to thank the anonymous reviewers for their careful reviews, insightful comments and very useful suggestions. This work was supported by the MEC/MINECO projects CONSOLIDER-INGENIO CSD2007-00022 and TIN 2010-21062-C02-02, GVA project PROMETEO/2008/051, the COST - European Cooperation in the field of Scientific and Technical Research IC0801 AT, and the *REFRAME* project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Ministerio de Economía y Competitividad in Spain.

References

Alonzo TA, Pepe MS, Lumley T (2003) Estimating disease prevalence in two-phase studies. Biostatistics 4(2):313–326

Anderson T (1962) On the distribution of the two-sample cramer-von Mises criterion. The Annals of Mathematical Statistics pp 1148–1159

Bakar AA, Othman ZA, Shuib NLM (2009) Building a new taxonomy for data discretization techniques. In: In Proc. 2nd Conference on Data Mining and Optimization (DMO'09), pp 132–140

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana M (2009a) Calibration of machine learning models. In: Handbook of Research on Machine Learning Applications, IGI Global

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana M (2009b) Similarity-binning averaging: A generalisation of binning calibration. In: International Conference on Intelligent Data Engineering and Automated Learning, LNCS, vol 5788, pp 341–349

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana M (2010) Quantification via probability estimators. In: International Conference on Data Mining, ICDM2010, pp 737–742

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana M (2012) On the effect of calibration in classifier combination. Applied Intelligence DOI 10.1007/s10489-012-0388-2

Chan Y, Ng H (2006) Estimating class priors in domain adaptation for word sense disambiguation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Associa-

tion for Computational Linguistics, Association for Computational Linguistics, pp 89–96
Chawla N, Japkowicz N, Kotcz A (2004) Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter 6(1):1–6
Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1–30
Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Prieditis A, Russell S (eds) Proceedings of the Twelfth International Conference on Machine Learning, pp 194–202
Ferri C, Hernández-Orallo J, Modroiu R (2009) An experimental comparison of performance measures for classification. Pattern Recognition Letters 30(1):27–38
Flach P (2012) Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press
Forman G (2005) Counting positives accurately despite inaccurate classification. In: Proceedings of the 16th European Conference on Machine Learning (ECML), pp 564–575
Forman G (2006) Quantifying trends accurately despite classifier error and class imbalance. In: Proceedings of the 12 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 157–166
Forman G (2008) Quantifying counts and costs via classification. Data Mining and Knowledge Discovery 17(2):164–206
Frank A, Asuncion A (2010) UCI machine learning repository. URL http://archive.ics.uci.edu/ml
González-Castro V, Alaiz-Rodríguez R, Alegre E (2012) Class distribution estimation based on the Hellinger distance. Information Sciences 218(1):146–164
Hastie TJ, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer
Hernández-Orallo J, Flach P, Ferri C (2012) A unified view of performance metrics: Translating threshold choice into expected classification loss. Journal of Machine Learning Research (JMLR) 13:2813–2869
Hodges J, Lehmann E (1963) Estimates of location based on rank tests. The Annals of Mathematical Statistics 34(5):598–611
Hosmer DW, Lemeshow S (2000) Applied logistic regression. Wiley
Hwang JN, Lay SR, Lippman A (1994) Nonparametric multivariate density estimation: a comparative study. Signal Processing, IEEE Transactions on 42(10):2795–2810
Hyndman RJ, Bashtannyk DM, Grunwald GK (1996) Estimating and visualizing conditional densities. Journal of Computational and Graphical Statistics 5(4):315–336
Moreno-Torres J, Raeder T, Alaiz-Rodríguez R, Chawla N, Herrera F (2012) A unifying view on dataset shift in classification. Pattern Recognition 45(1):521530

Neyman J (1938) Contribution to the theory of sampling human populations. Journal of the American Statistical Association 33(201):101–116

Platt JC (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, MIT Press, pp 61–74

R Team, et al (2012) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Raeder T, Forman G, Chawla N (2012) Learning from imbalanced data: Evaluation matters. Data Mining: Foundations and Intelligent Paradigms 23:315–331

Sánchez L, González V, Alegre E, Alaiz R (2008) Classification and quantification based on image analysis for sperm samples with uncertain damaged/intact cell proportions. In: Proceedings of the 5th International Conference on Image Analysis and Recognition, Springer, LNCS, vol 5112, pp 827–836

Sturges H (1926) The choice of a class interval. J American Statistical Association 21(153):65–66

Tenenbein A (1970) A double sampling scheme for estimating from binomial data with misclassifications. Journal of the American Statistical Association 65(331):1350–1361

Weiss G (2004) Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1):7–19

Weiss G, Provost F (2001) The effect of class distribution on classifier learning: an empirical study. Technical Report ML-TR-44

Witten IH, Frank E (2005) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Elsevier

Xiao Y, Gordon A, Yakovlev A (2006a) A C++ program for the Cramér-von Mises two-sample test. Journal of Statistical Software 17:1–15

Xiao Y, Gordon A, Yakovlev A (2006b) The L1-version of the Cramér-von Mises test for two-sample comparisons in microarray data analysis. EURASIP Journal on Bioinformatics and Systems Biology pp 7–7

Xue J, Weiss G (2009) Quantification and semi-supervised classification methods for handling changes in class distribution. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 897–906

Yang Y (2003) Discretization for naive-bayes learning. PhD thesis, Monash University

Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: Proceedings of the 8th International Conference on Machine Learning (ICML), pp 609–616

Zadrozny B, Elkan C (2002) Transforming classifier scores into accurate multiclass probability estimates. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 694–699