Prog Artif Intell
DOI 10.1007/s13748-016-0103-3

REGULAR PAPER

Why is quantification an interesting learning problem?

Pablo González[1] · **Jorge Díez**[1] · **Nitesh Chawla**[2,3] · **Juan José del Coz**[1,2]

Received: 11 August 2016 / Accepted: 19 September 2016

Abstract There are real applications that do not demand to classify or to make predictions about individual objects, but to estimate some magnitude about a group of them. For instance, one of these cases happens in sentiment analysis and opinion mining. Some applications require to classify opinions as positives or negatives, but there are also others, even more useful sometimes, that just need an estimation of which is the proportion of each class during a concrete period of time. "How many tweets about our new product were positive yesterday?" Practitioners should apply quantification algorithms to tackle this kind of problems, instead of just using off-the-shelf classification methods, because classifiers are suboptimal in the context of quantification tasks. Unfortunately, quantification learning is still relatively an under explored area in machine learning. The goal of this paper is to show that quantification learning is an interesting open problem. To support its benefits, we shall show an application to analyze Twitter comments in which even the most simple quantification methods outperform classification approaches.

Keywords Sentiment analysis · Opinion mining · Quantification · Prevalence estimation · Population shift

✉ Juan José del Coz
juanjo@aic.uniovi.es; juanjodelcoz@gmail.com;
juanjo@uniovi.es

Pablo González
pglez82@gmail.com

Jorge Díez
jdiez@uniovi.es

Nitesh Chawla
nchawla@nd.edu

1 Artificial Intelligence Center, University of Oviedo at Gijón, Oviedo, Spain

2 Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

3 Interdisciplinary Center for Network Science and Applications, University of Notre Dame, Notre Dame, IN 46556, USA

1 Introduction

Several real-world problems demand effective automatic methods to estimate the class distribution of a sample. For instance, to measure the success of a new product, companies are interested on the overall consumers' opinion and less about an individual consumer opinion. To answer questions like *how many clients are satisfied with our new product?*, we need effective algorithms focused on estimating the distribution of the classes. We can find other examples in problems aimed at tracking trends over time [17], such as market and ecosystems evolution [3], portfolio credit risk [19], collective behavior in social networks [15], monitoring of support-call logs [9] or any other kind of distribution summarization in general. In these cases, it is sufficient to obtain estimations of the distribution to properly plan strategies or policies. Such global estimation is more relevant than classifying individual examples, especially when the number of such individual predictions is huge, hindering their posterior analysis.

This learning problem, named quantification [8], class probability estimation [12], or prevalence estimation [13] depending on the author, does not produce/require individual predictions, but aggregated ones. Thus, quantification learning represents a new problem different than other supervised learning tasks, like classification. First, the goal of quantification learning is distinctly different than classification. While in supervised classification, the aim is to learn a model:

$$h : \mathcal{X} \longrightarrow \{c_1, \ldots, c_l\}, \tag{1}$$

where $\mathcal{X}$ is the input space to represent the individual examples and $\{c_1, \ldots, c_l\}$ are the set of l possible classes, in quantification that the learning task is:

$$\bar{h} : \mathcal{X}^n \longrightarrow [0, 1]^l, \tag{2}$$

that is, given a set of instances, the model $\bar{h}$ must return the probability distribution of the classes. Obviously, the performance measures also differ. While in classification tasks, one generally leverages evaluation metrics such as accuracy, AUC, in quantification learning, one is more interested in comparing probability distributions or measuring the counts outcome via mean absolute error or mean squared error as done in regression.

Thus, in essence quantification, problems are quite different than those that follow the classical classification setting. Not only because the final goal is different, but also, and more importantly, because the learning assumptions are totally different. In classification learning, it is generally assumed that training data are independent and identically distributed (i.i.d.), while this is not true by definition in the case of quantification tasks. Maybe the most important property of quantification learning is that the distribution of data changes between training and testing phases. This is due to the task definition itself. Ignoring this last characteristic will lead to obtain poor models, or at least suboptimal solutions.

Despite these pressing applications of quantification learning, they generally remain under-explored. This is probably due to several and different reasons. First, because practitioners think that quantifying is not a difficult task, it seems as simple as classifying and counting examples. If we build a "good" classifier, then we will obtain "good" prevalence estimations. However, there are better alternatives, algorithms that are more precise, stable and robust than the *Classify & Count* (CC) approach. However, the main reason for quantification learning being ignored is that the majority of experts in machine learning do not focus on quantification. For instance, quantification methods are not included in any publicly available machine learning library. Under such circumstances, it is hardly surprising that quantifiers are not being used as much as they should.

The goals of this paper are twofold. First, to show that quantification is not a byproduct of classification. Quantification represents a different learning task that demands its own learning methods. The main principles of quantification learning will be described and we shall justify why the classify and count approach is an inappropriate solution. Second, we shall show a concrete application, aimed at quantifying Twitter comments, in which easy-to-implement quantification techniques are much better than CC.

2 Quantification learning

To better understand quantification learning, it is important to establish its basic ideas. Fawcett and Flach [7] proposed a taxonomy to classify learning problems according to the causal relationship between class labels and covariates. The interest of this taxonomy is that it determines the kind of changes in the distribution that a particular task can suffer from. They distinguished two different kinds of problems: $\mathcal{X} \rightarrow \mathcal{Y}$ in which the class label is causally determined by the values of the covariates and problems $\mathcal{Y} \rightarrow \mathcal{X}$ where the class label causally determines the covariates. The joint distribution of both $P(x, y)$ can be written as $P(y|x)P(x)$ in $\mathcal{X} \rightarrow \mathcal{Y}$ problems and $P(x|y)P(y)$ in the case of problems $\mathcal{Y} \rightarrow \mathcal{X}$. Quantification falls in the second category, and its main learning assumptions are that: (1) $P(y)$ changes over the time (this is the motivation of the problem), but (2) $P(x|y)$ remains constant. As we shall see, it is important to keep this in mind to understand the ideas of the quantification learners described below.

If we restrict ourselves to the simplest case, binary quantification ($l = 2$), we have a training set with examples labeled as positives or negatives; formally, $D = \{(x_i, y_i) : i = 1 \ldots n\}$, in which x_i is an object of the input space $\mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$. This data set shows a specific distribution that can be summarized with the actual proportion of positives or prevalence p. The learning goal is to obtain a model able to predict the prevalence ($p' = P(y = +1)$) of another unlabeled sample that may show a markedly different distribution of the classes. Thus, the input data are equivalent to that of the traditional classification problems, but the focus is on the estimated prevalence (p') of the sample, rather than on the class assigned to each individual example. Notice that we use p and p' to identify the actual and estimated prevalences of any sample.

3 Count and adjust

Maybe the most relevant quantification method is the adjusted count (AC) algorithm proposed by Forman [8]. Not only because it was one of the first methods proposed, but especially because it is theoretically well founded and justified why quantification algorithms are needed. The main idea behind AC is to correct the estimate provided by a classifier, depending on its expected behavior characterized by its true positive rate (*tpr*) and false positive rate (*fpr*). AC follows a two-step procedure, both during the learning phase and when predictions are made. To obtain the quantifier, first AC trains a classifier and then estimates its $tpr = TP/(TP + FN)$ and $fpr = FP/(TN + FP)$ by means of a cross-validation over the training set. To predict the prevalence of a given unlabeled sample, AC counts the positive predictions of the classifier over the examples in the sample (i.e., like the CC method described previously) obtaining a first estimate p'_0, which is afterwards corrected using tpr and fpr values.

This correction is based on the fact that the obtained estimate p'_0 can be expressed in function of the actual prevalence p by means of tpr and fpr:

$$p'_0(p) = p \cdot tpr + (1 - p) \cdot fpr. \tag{3}$$

That is, only the *tpr* fraction of the positives (p) will be perceived as true positives by the classifier ($tpr \cdot p$), but at the same time, the *fpr* fraction of the negatives ($1 - p$) will be misclassified as false positives ($fpr \cdot (1 - p)$). The sum of both terms gives us the value p'_0.

The first result of this relationship is that a better estimate can be obtained given p'_0, tpr, and fpr, just isolating p from the previous equation and computing its value to obtain the proportion of positive examples p' finally returned by AC:

$$p' = \frac{p'_0 - fpr}{tpr - fpr}. \tag{4}$$

Notice that theoretically AC should return perfect estimates p', independently of the quality of the underlying classifier, whenever $P(x|y)$ remains constant and we perfectly estimate tpr and fpr values. If the within-class probability densities, $P(x|y)$, do not change, this in turn ensures that both classifier characteristics, *tpr* and *fpr*, are independent of changes in class distribution, see Fawcett and Flach [7]. However, in practice, it happens that one of these two aspects is not true. Usually, the estimates of tpr and fpr are not perfect, and maybe there are some variations in $P(x|y)$. Despite these aspects, AC improves the performance of CC approach in all the studies published in the literature.

4 Why is classify and count a bad quantifier?

A second result that can be obtained from the relationship between the prevalence estimation of a classifier and the actual prevalence (Eq. 3) is that we can prove why CC approach usually performs poorly. Of course, there is one exception: if a perfect classifier can be obtained, then CC will also be a perfect quantifier. However, for the rest of situations, which is the usual case in a real-world application, the performance of CC approach tends to be poor, especially when the distribution of the classes changes significantly.

The demonstration[1] is based on the idea that CC will give a perfect estimate for a particular prevalence, denoted by p^*, and then, we can study what happens when such prevalence changes. If CC correctly estimates the prevalence for p^*, i.e., $p'_0(p^*) = p^*$, then for a different prevalence $p^* + \Delta$ with $\Delta \neq 0$, CC will not predict the correct prevalence:

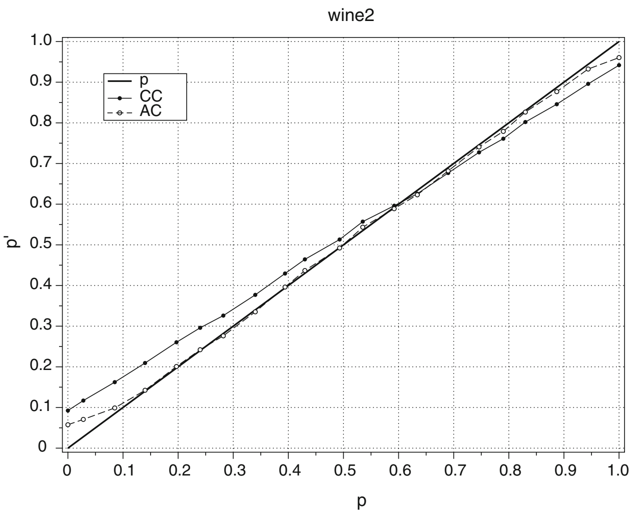

Fig. 1 Analysis of bias when the prevalence varies in [0:0.05:1]. The diagonal represents the perfect quantifier. The figure compares *CC* with the *AC* method. Each result is the average of 100 sample sets with the same prevalence. The bias of *CC* shows the expected behavior

$$\begin{aligned} p'_0(p^* + \Delta) &= tpr \cdot (p^* + \Delta) + fpr \cdot (1 - (p^* + \Delta)) \\ &= p'_0(p^*) + (tpr - fpr) \cdot \Delta \\ &= p^* + (tpr - fpr) \cdot \Delta. \end{aligned}$$

Notice that $p'_0(p^* + \Delta) = p^* + \Delta$ only if $tpr - fpr = 1$, which means that the underlying classifier is perfect ($tpr = 1$ and $fpr = 0$). Since usually the classifier is imperfect, $tpr - fpr < 1$, if Δ is positive, the estimate p'_0 is less than $p^* + \Delta$, and if Δ is negative, the estimate is greater than $p^* + \Delta$. An example of this behavior is shown in Fig. 1. The CC method correctly estimates the prevalence for $p \approx 0.58$, then CC underestimates the prevalence when it increases, and overestimates it when decreases.

5 Other quantification methods

Despite quantification learning not having been deeply studied yet, there are already a significant number of quantification methods. Forman proposed several algorithms based on threshold selection policies. These methods are based on training a linear SVM classifier and to employ then a posterior calibration of its threshold. The main difference between these methods is the threshold selection policy applied, aimed at alleviating some drawbacks of AC correcting formula (Eq. (4)) for certain cases. A key problem related to the AC method is that its performance may depend on the degree of imbalance in the training set, worsening when the positive

[1] All the details can be found in [8]

class is scarce. In such case, the underlying classifier tends to minimize the false positive errors, which usually implies a low *tpr* and a small denominator in Eq. (4). This fact produces high vulnerability to fluctuations in the estimation of *tpr* or *fpr*. For highly imbalanced situations, the main intuition is that selecting a threshold that allows more true positives, even at the cost of many more false positives, can afford a better quantification performance. The goal is to choose those thresholds where the estimates of *tpr* and *fpr* have less variance or where the denominator in Eq. (4) is large enough to be more resistant to estimation errors. As stated previously, the performance of AC mainly depends on the quality of tpr and fpr estimations. Therefore, several policies can be considered, for instance, the X method chooses the threshold where *fpr* equals $1 - tpr$, avoiding the tails of both curves. Notwithstanding, the estimation of *tpr* and *fpr* can still differ significantly from the real values. Forman thus proposed a more advanced method, *Median Sweep* (MS), based on estimating the prevalence for all possible thresholds, to compute their median. The drawback of this approach is that several estimations of tpr and fpr are required, one for each threshold.

In [4], a probabilistic version of AC is developed. First, the authors introduce a simple method called *Probability Average* (PA), which is clearly aligned with CC. The key difference is that the classifier learned is probabilistic and the prevalence returned is computed as the average of the probabilities for the positive class. As it might be expected, when the proportion of positives changes between training and testing, then PA will underestimate or overestimate it as CC does. These authors thus propose an enhanced version of this method, called *Scaled Probability Average* (SPA). Similar to CC and AC, the first estimation obtained by PA is corrected according to a simple scaling formula.

Finally, there are quantifiers based on traditional learning methods, like instance-based learning [1] or decision trees [14], but also using more recent approaches, like ensembles Pérez-Gallego et al [16] and structured output learning [2,5,6]. In particular, [2] presents a method, called Q, based on building a classifier that optimizes a loss function (*Q-measure*), inspired in the popular *F-measure*, that combines the classification and the quantification performance of the model through a parameter β. One difficulty in implementing this idea is that not all binary learners are capable of optimizing this kind of metric, because such loss functions are not decomposable as a linear combination of the individual errors. Hence, this approach requires a multivariate prediction learning machine. However, the straightforward benefit is that Q method addresses the quantification task from an aggregated perspective, considering the performance over complete samples, which seems more appropriate for the problem in general.

6 Tracking consumers' opinion

To prove the benefits of using quantification learning, we tested several quantification algorithms in the context of sentiment analysis. Two different data sets were employed. The first one, Sentiment140 [11], is composed by $1,600,000$ Twitter messages with emoticons collected in the time period between April 6, 2009 to June 16, 2009. The tweets were labeled using distant supervision, that is, the emoticons serve as noisy labels. For example, :-) in a tweet indicates that it contains a positive comment, while :-(indicates that the tweet contains a negative sentiment. The emoticons were used only for labeling purposes and were removed from the description of the training examples afterwards.

The nice aspect of this data set is that it allows us to perform a quite realistic experiment. We can train a quantification model with the tweets form the first day and then try to predict the prevalence of positive comments for the rest of the period, tracking somehow consumers' opinion. The results of such experiment are shown in Fig. 2. To analyze the results, it is important to consider that CC, AC, X, and MS methods all use the same classifier, the differences between them are the threshold applied (for X and MS), and the correction made using Eq. 4.

As we can observe, the estimations of CC are much worse than those of proper quantification algorithms. Actually, the problem is not very difficult from a quantification perspective, because the prevalence of positive comments is quite

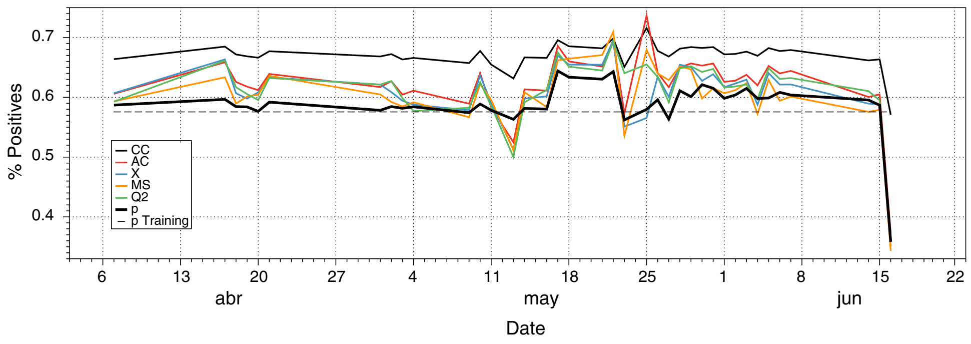

Fig. 2 Results for Sentiment140 data set. The figure shows the actual prevalence p and the proportion estimated p' by each quantification method for each day. The prevalence of the training data is showed with a *dotted line*

constant and similar to the one of the training data; let us recall that those are the perfect conditions for CC. Despite this fact, the performance of CC can be easily improved using any quantification method, for instance, methods X and Q (with $\beta = 2$) offer better estimations than CC for all the days of the period, while AC and MS output a worse prediction than CC for just one day each. Notice that for the day with the greatest shift in the prevalence of positive opinions (the last day it drops to 0.35), the prediction made by CC is quite poor, showing the kind of behavior that was discussed before in Fig. 2.

Table 1 Absolute error for each method. The last rows show the average and the average ranking

Date	*CC*	*AC*	X	MS	Q2
7-apr	7.6800	1.8900	1.9800	0.6200	0.5673
17-apr	8.8200	6.2000	6.6800	3.6900	6.4657
18-apr	8.7300	4.1300	2.2300	0.5100	3.2485
19-apr	8.4600	3.3800	1.3900	1.7300	2.1428
20-apr	8.9800	3.5500	3.1600	2.7900	1.8838
21-apr	8.5100	4.7200	4.1300	4.4000	4.0410
1-may	9.0100	3.8900	4.3100	2.6500	4.3069
2-may	8.7700	4.2400	2.4200	0.7700	4.2661
3-may	8.1600	2.2800	1.2500	0.3000	1.6982
4-may	8.1800	2.6900	0.3600	0.7600	0.7178
9-may	8.3000	1.5100	0.4100	0.7500	0.8277
10-may	8.8800	5.1700	4.7800	3.3100	3.6506
11-may	7.6400	0.4800	1.2100	1.7400	0.7168
13-may	6.8400	3.8500	5.0500	5.1900	6.3184
14-may	8.5400	3.1900	1.6400	2.7000	1.0217
16-may	8.5600	3.0900	2.1200	0.3300	3.2126
17-may	5.1600	4.1800	2.5700	1.7800	3.0788
18-may	5.1700	2.6400	2.1000	3.0900	1.7585
21-may	5.1600	2.1100	2.4100	4.0300	1.4508
22-may	5.5400	4.9800	5.2500	6.6500	5.0246
23-may	8.8800	1.1200	1.1200	2.6800	7.8177
25-may	13.6100	15.7100	1.4400	9.9800	7.5200
26-may	8.2000	4.5100	3.8800	4.4100	3.8420
27-may	10.4600	5.3300	3.7200	6.5300	2.7944
28-may	7.0400	3.8900	4.3500	4.0800	3.7617
29-may	8.3000	5.5600	4.9500	4.5600	5.0853
30-may	6.1500	3.1900	0.6400	2.3200	2.1324
31-may	6.8800	4.1400	2.3500	0.0200	3.2249
1-jun	7.3500	2.7600	1.8700	0.8400	1.7843
2-jun	6.8600	2.3900	1.9100	0.8100	1.4224
3-jun	6.1700	2.2900	1.4800	0.8900	0.6697
4-jun	7.1400	2.2000	1.0300	2.6000	0.1329
5-jun	8.3500	5.3600	4.1500	3.0300	4.9808
6-jun	6.9100	3.1700	1.3100	1.4100	2.2472
7-jun	7.5200	4.0400	1.7600	0.2600	2.8390
14-jun	6.6200	0.5100	0.6100	1.9800	1.4863
15-jun	7.6700	1.8200	0.1400	0.7600	0.8091
16-jun	21.2500	1.3800	0.8200	1.5400	0.1555
Avg.	8.1697	3.6195	2.4468	2.5392	2.8712
Avg. rank	4.9474	3.2500	2.2237	2.3158	2.2632

Table 2 Mean absolute error for each quantification method and prevalence over the STS-gold data set. In the bottom, the average ranking

p	CC	AC	X	MS	Q2
0.0	12.6200	0.8955	0.2703	1.1070	0.5787
0.1	8.1310	1.8650	2.5010	3.7200	2.2729
0.2	3.8240	2.7980	3.4780	4.2850	2.8929
0.3	1.0990	2.1970	2.9040	3.8970	2.3139
0.4	4.5480	3.1120	3.3400	3.9010	3.8717
0.5	8.8600	4.0770	3.5630	5.7000	4.8445
0.6	14.0400	3.6880	5.0650	2.7770	4.1483
0.7	17.4100	4.2180	5.7480	4.0840	5.7662
0.8	21.9700	4.1000	5.6820	3.7230	6.3985
0.9	25.5000	4.2690	4.8030	4.8200	5.9561
1.0	30.3800	2.3570	4.5260	1.4990	3.0486
Avg.	13.4893	3.0524	3.8073	3.5921	3.8266
Avg. rank	4.5455	1.7273	2.7273	3.0000	3.0000

The differences between AC and CC are statistically significant applying a Friedman-Nemenyi test ($p = 0.05$)-value

The same results are shown numerically in Table 1. In the last row, we can see the average ranking of each algorithm. It is remarkable that all proper quantifiers (AC, X, MS, and Q2) are significantly better ($p - value = 0.01$) than CC using a Friedman–Nemenyi test Garcia and Herrera [10].

The second data set, taken from [18], has 2034 tweets labeled manually by three annotators that agreed on their sentiment labels (positive/negative). The positive class has a prevalence of 0.31. This experiment was less realistic, because the data set is smaller and does not contain any temporal information, thus we performed a typical cross validation experiment. For each testing fold, we generated 11 different samples, with the prevalence varying from 0.0 to 1.0. Table 2 shows the *mean absolute error* (MAE) scores for each prevalence. Again the performance of CC is worse than that of quantification methods. Noteworthy, the results of CC are very good when the prevalence (0.3) is similar to the one observed in the training set, but they are rather poor when the prevalence is significantly different. The worse results are in the two extremes.

7 Conclusions

Quantification is an interesting learning task not only because theoretically presents challenging properties from a conceptual point of view, but also because it can be applied to tackle quite important real-world problems. Unfortunately, we do think that quantification learning has not received the attention that deserves from the community. We hope that in the near future, more researchers will contribute to this field and quantification methods will be applied for those applications that require to estimate the class distribution of samples.

Acknowledgments This research has been funded by MINECO (the Spanish Ministerio de Economía y Competitividad) and FEDER (Fondo Europeo de Desarrollo Regional), Grant TIN2015-65069-C2-2-R. Juan José del Coz is also supported by the Fulbright Commission and the Salvador de Madariaga Program, Grant PRX15/00607. This paper has been written during the stay of Juan José del Coz at the University of Notre Dame.

References

1. Barranquero, J., González, P., Díez, J., del Coz, J.J.: On the study of nearest neighbour algorithms for prevalence estimation in binary problems. Pattern Recognit. **46**(2), 472–482 (2013)
2. Barranquero, J., Díez, J., del Coz, J.J.: Quantification-oriented learning based on reliable classifiers. Pattern Recognit. **48**(2), 591–604 (2015)
3. Beijbom, O., Hoffman, J., Yao, E., Darrell, T., Rodriguez-Ramirez, A., Gonzalez-Rivero, M., Guldberg, O.H.: Quantification in-the-wild: data-sets and baselines. In: NIPS 2015, Workshop on Transfer and Multi-Task Learning. Montreal, CA (2015)
4. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.: Quantification via probability estimators. In: Proc. of the 10th IEEE International Conference on Data Mining, pp. 737–742 (2010)
5. Esuli, A., Sebastiani, F.: Sentiment quantification. IEEE Intell. Syst. **25**(4), 72–75 (2010)
6. Esuli, A., Sebastiani, F.: Optimizing text quantifiers for multivariate loss functions. ACM Trans. Knowl. Discov. Data **9**(4), 27:1–27:27 (2015)
7. Fawcett, T., Flach, P.: A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. Mach. Learn. **58**(1), 33–38 (2005)
8. Forman, G.: Quantifying counts and costs via classification. Data Mining Knowl. Discov. **17**(2), 164–206 (2008)
9. Forman, G., Kirshenbaum, E., Suermondt, J.: Pragmatic text mining: minimizing human effort to quantify many issues in call logs. In: Proceedings of ACM SIGKDD'06, ACM, pp. 852–861 (2006)
10. Garcia, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J. Mach. Learn. Res. **9**, 2677–2694 (2008)
11. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1:12 (2009)
12. González-Castro, V., Alaiz-Rodríguez, R., Alegre, E.: Class distribution estimation based on the hellinger distance. Inf. Sci. **218**, 146–164 (2013)
13. Latinne, P., Saerens, M., Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities may significantly improve classification accuracy: Evidence from a multi-class problem in remote sensing. In: Proceedings of ICML'01, M. Kaufmann, pp. 298–305 (2001)
14. Milli, L., Monreale, A., Rossetti, G., Giannotti, F., Pedreschi, D., Sebastiani, F.: Quantification trees. In: IEEE International Conference on Data Mining (ICDM'13), pp. 528–536 (2013)
15. Milli, L., Monreale, A., Rossetti, G., Pedreschi, D., Giannotti, F., Sebastiani, F.: Quantification in social networks. In: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, pp. 1–10 (2015)
16. Pérez-Gallego, P., Quevedo, J.R., del Coz, J.J.: Using ensembles for problems with characterizable changes in data distribution: a case study on quantification. Inf. Fusion **34**, 87–100 (2017)
17. Rakthanmanon, T., Keogh, E., Lonardi, S., Evans, S.: MDL-based time series clustering. Knowl. Inf. Syst. **33**(2), 371–399 (2012)
18. Saif, H., Fernández, M., He, Y., Alani, H.: Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. In: 1st Interantional Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013) (2013)
19. Tasche, D.: Exact fit of simple finite mixture models. J. Risk Financial Manag. **7**(4), 150–164 (2014)