Dynamic Reliable Voting in Ensemble Learning

Agus Budi Raharjo, Mohamed Quafafou

Dynamic Reliable Voting in Ensemble Learning

Agus Budi Raharjo and Mohamed Quafafou

Aix Marseille University, Université de Toulon, CNRS, LIS, Marseille, France
{agus-budi.raharjo, mohamed.quafafou}@univ-amu.fr

Abstract. The combination of multiple classifiers can produce an optimal solution than relying on the single learner. However, it is difficult to select the reliable learning algorithms when they have contrasted performances. In this paper, the combination of the supervised learning algorithms is proposed to provide the best decision. Our method transforms a classifier score of training data into a reliable score. Then, a set of reliable candidates is determined through static and dynamic selection. The experimental result of eight datasets shows that our algorithm gives a better average accuracy score compared to the results of the other ensemble methods and the base classifiers.

Keywords: confidence score · reliable voting · ensemble learning

1 Introduction

Nowadays, data is available from heterogeneous and dynamic web data sources and their integration becomes a crucial problem [5]. Due to the increase of such data sources and/or data Web Services, finding and ranking the suitable data sources and/or data web services leads to deep investigations and significant research efforts [12]. But from another viewpoint, processing the collected heterogeneous data remains a challenging problem especially for classification tasks. The objective of multiple classifiers is to build a powerful solution to handle a difficult pattern recognition problems [2]. There are two existing categories to combine classifiers, i.e. weighting and meta-learning [11]. Both categories tend to focus on the label prediction to build a decision than considering the class probability. Predicted class probability can be used to represent the classifier's confidence score as it is applied in weighted voting [9]. However, the quality of the weighted voting depends on the performance of its base classifiers. Moreover, each supervised learning has its own method to provide a confidence score and it does not always in probability form. For these reasons, we extend the previous study on the transformation of confidence score [15]. In the context of ensemble learning, each base learner has its own performance. The reliability score of each classifier within the binary problem is considered to handle the contrasted performances and to propose a better prediction. It is important to mention that prediction score also represents algorithm's level of certainty for each instance of a dataset, so that the composition of base classifiers can be changed dynamically based on their confidence level. We propose a new weighted voting approach that adapts to varied data characteristics. The reliability aspect is tested by adding spammers to base classifiers. Detailed explanation of this paper is organized as follows: section 2 reviews the previous works on supervised learning and ensemble methods; the proposed algorithm is described in section 3; section 4 is dedicated to the experiment setup, result, and discussion; and concluding remarks are given in section 5.

2 Related Work

In this section, a brief description about several supervised learning approaches is provided. These approaches are used as base classifiers. Then, different combinations of ensemble method are discussed as previous works.

2.1 Supervised Learning as Base Classifiers

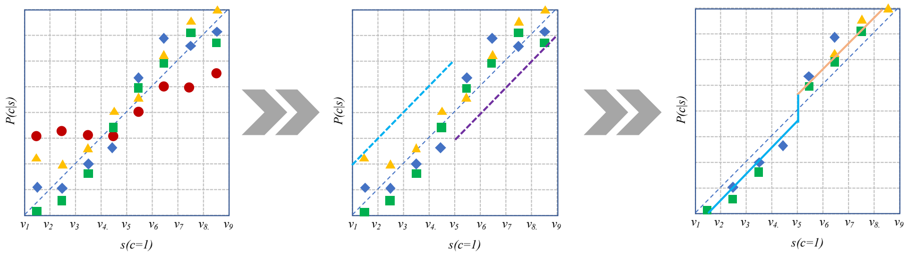
There are five approaches to represent a knowledge in supervised learning, Bayesian classifier, decision tree, rule, function, and lazy classifier [14]. Bayesian is classification method developed from Bayes' rule of conditional probability. Decision tree is a "divide-and-conquer" approach in learning problem from a set of independent instances with attribute as its node. Rule has similar representation as decision tree, although several preconditions are applied to determine its conclusion. Function learner consists of various algorithms that can be written down as mathematical equations in a reasonably natural way. Lazy works by delaying the classification until all the training data is collected and a request is made. In addition to the above algorithms, there is a predictor called spammers who label a class randomly, which are often found in the case of crowds [10]. One algorithm of each category and a group of spammers are selected as ensemble input to ensure the diversity of base classifier, which is further discussed in section 4.

2.2 Confidence Score in Ensemble Learning

Voting, Stacked Generalization (Stacking), and Multi-Scheme are some examples of combination method that are able to handle different base classifiers as the inputs, while Random Forest, Bootstrap aggregating (Bagging), and Boosting combine several models from the same algorithm [8]. Majority voting (MV) is a popular combination method compared to the others [16]. The quality of MV depends on the performance of base classifiers. In order to make a robust voting, weighted voting (WMV) can be considered as a potential method [13]. Confidence score can be used as weight parameter of voting. This method is conducted by collecting confidence scores from a set of classifiers and take the average of each class. The highest value determines which label belongs to the tested data. Our work enhances the utilization of confidence score to build a reliability diagram. Our proposed method is compared to MV, WMV, Stacking, and Multi-Scheme, since several approaches of classifiers are used as the basis.

3 Dynamic Reliable Voting Algorithm

In this section, we specify our problem and basic notation used in binary classification. We also provide an overview of reliability diagram according to the previous literatures. Then, the workflow of Dynamic Reliable Voting (DRV) algorithm is proposed. As shown in Figure 1, our approach can be summarized into three steps: (a) transforming confidence score into reliability diagram, (b) removing spammer or weak classifier by static threshold, (c) selecting the best combination decision for each bin. X-axis represents the confidence score, while Y-axis is the value of empirical class membership probability. Four different points in Figure 1a describe the diversity of learning algorithms. Static thresholds are illustrated as blue and purple dash lines (see Fig. 1b), and dynamic threshold is drawn in a solid blue-orange line (see Fig. 1c).



(a) Transforming confidence score into reliability diagram

(b) Static classifier selection

(c) Dynamic classifier selection

Fig. 1. The framework of the proposed algorithm

3.1 Problem Formulation and Modeling

The training process of ensemble learning is started by a prediction of a set of instances $X = \{x_1, x_2, ..., x_i, ..., x_N\}$ by T base classifiers. Then, the decisions from multiple models are combined to improve the overall performance. Each decision of classifier t consists of an independent label prediction y and its confidence score s. This ensemble decisions can be noted as follow: $D = \big\{(y_i^1, s_i^1), (y_i^2, s_i^2), ..., (y_i^t, s_i^t), ..., (y_i^T, s_i^T)\big\}_{i=1}^{N}$ such that $y_i^t \in \{0, 1\}$, $s_i^t \in [0, 1]$, and $s_i^t \in \mathbb{R}$. A confidence score needs to be normalized if $s \in [a, b]$, where $a < b$, $a \neq 0$ or $b \neq 1$, $a \in \mathbb{R}$, and $b \in \mathbb{R}$. The normalization can be expressed as follow: $s' = \frac{s-a}{b-a}$. In a case where a and b are unknown, Platt scaling [6] can be applied to adjust the confidence score range.

Reliability diagram is firstly introduced to display forecast probability at Chicago [4]. This probability and observed relative frequency are drawn into X-axis and Y-axis diagram, respectively. Later, this representation is applied in classification by converting a confidence score into empirical class membership probability. This probability can be denoted as $P(c|s)$, where $c \in \{0, 1\}$ and c is a class label. This diagram is built in training step because it requires the true label of each instance, which is denoted as z_i and $z_i = \{0, 1\}$. In order to reflect the distribution of s, confidence score is converted according to the class c, which is $s(c = 1)^t_i = 1 - s(c = 0)^t_i$ for binary case. Then, a set of $s(c = 1)$ is split into several bins in an interval v. Let V be a set of interval, then $V = \{v_j | v_{j+1} - v_j = v_j - v_{j-1}, v_j \in \mathbb{R}, v_j \in [0, 1 + \Delta v[\}$. s^t_i will be categorized in interval j when $v_j \leq s^t_i < v_{j+1}$. $P(c|s)$ is defined as the number of true label corresponded to c divided by the number of all prediction in interval j. The result of the distribution is represented in Figure 1a.

3.2 Classifier Selection

Once we get the information of reliability representation for each classifier, a threshold is defined to filter reliable classifiers, which is denoted as RC. The selection process contains two step, static and dynamic selection. In static phase, the average probability estimate $\overline{P(c|s^t_i)}$ and the training accuracy A^t of each classifier are calculated, which is formulated by equation 1. A set of reliable candidate RC is determined by choosing algorithms that satisfy the threshold ϵ_1 and ϵ_2, or have an accuracy A^t better than the average accuracy $\overline{A}$. The purpose of this step is to eliminate a probable spammer in the training step, since the $\overline{P(c|s^t_i)}$ of spammer lies in the area of uncertainty (from 0.4 to 0.6).

$$RC = \left\{ t \in T | \left(\overline{P(c|s^t_{1i})} \leq \epsilon_1 \wedge \overline{P(c|s^t_{2i})} \geq \epsilon_2 \right) \vee \left(A^t > \overline{A} \right) \right\} \tag{1}$$

where $s^t_{1i} = s^t_i \leq 0.5$, $s^t_{2i} = s^t_i > 0.5$, $\epsilon_1 \in \mathbb{R}$, $\epsilon_1 \in [0, 0.5]$, $\epsilon_2 \in \mathbb{R}$, and $\epsilon_2 \in [0.5, 1]$.

Dynamic selection process is started by searching the optimal values of thresholds $\lambda = \{\lambda_1, \lambda_2\}$, where $\lambda \in \mathbb{R}$ and $\lambda \in [-0.5, 0.5]$. This selection is applied in the testing step. A classifier will be excluded from RC if the confidence score is in the range of $[0, 0.5]$ and its probability estimate is higher than the bin threshold, while a classifier which has a confidence score between 0.51 to 1 will be eliminated from RC if its probability estimate is less than the bin threshold (see eq. 2). The final reliable classifier RC_f is defined as a subset of RC, which contains a set classifiers that pass the threshold. This selection is called a dynamic process because of the value of confidence score for each instance s^t_i is different and independent, so the number of RC_f is also different for each test datum.

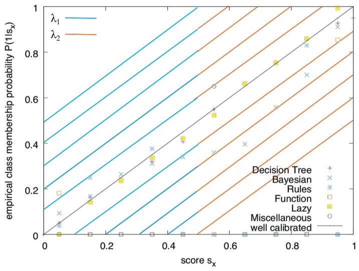


Fig. 2. An example of the reliability diagram of six classifiers with the possible thresholds of λ_1 and λ_2 by the precision 0.1.

$$RC_f = \begin{cases} P(c|s^t_i) \leq ((v_j + v_j + 1)/2) - \lambda_1 & \text{if } s^t_i \leq 0.5 \\ P(c|s^t_i) \geq ((v_j + v_j + 1)/2) - \lambda_2 & \text{otherwise} \end{cases} \tag{2}$$

There is no efficient way to define the values of $\epsilon_1, \epsilon_2, \lambda_1$, and λ_2 except with an iterative process. The time complexity of the iterative loop C depends on its precision p, which can be written as follows:

$$C = \Big(\frac{0.6 - 0.4}{p}\Big)^2 \Big(\frac{0.5 - (-0.5)}{p}\Big)^2 \tag{3}$$

where $C \in \mathbb{R}$, $p \in \mathbb{R}$ and $p \in [0, 1]$. 0.6 and 0.4 are the highest and lowest limit of uncertainty respectively, while 0.5 and -0.5 are the the highest and lowest limit of λ respectively. The representation of the possible values of λ is illustrated in Figure 2. C can be optimized by limiting the values of λ_1 and λ_2 so that $0 < ((v_j + v_j + 1)/2) - \lambda_1 \leq 0.5$ and $0.5 < ((v_j + v_j + 1)/2) - \lambda_2 \leq 1$.

Our proposed method can be explained through algorithm 1. Line 1 describes the instances X that consists of X_{train} and X_{test}, the ground truth of the training data Z_{train}, base classifiers T, and a set if the interval limit V. The training step of base classifiers is processed in lines 2-4. Reliability transformation and static selection are conducted in lines 5-16. Then, lines 17-19 determine the optimal thresholds of λ_1 and λ_2. After defining the reliable candidate RC and the thresholds, a set of reliable candidate RC_f is extracted from the process in line 20 - 23. As it is shown in the line 20, the combination of RC_f depends on the characteristic of each instance x, hence it is called dynamic selection. Finally, a majority voting is applied in line 23 to produce a set of recommended decision.

Algorithm 1 Dynamic Reliable Voting

```
 1: Input: $X$, $Z_{train}$, $T$, interval limit $V$
 2: for $t \in T$ do
 3:     Build classifier $t = f(X_{train}, Z_{train})$
 4:     Calculate the accuracy $A$
 5: $\overline{A}_{max} \leftarrow 0$
 6: for $\epsilon_1 \in [0, 0.5]$ and $\epsilon_2 \in [0.5, 1]$ do
 7:     $RC \leftarrow \emptyset$
 8:     for $t \in T$ do
 9:         Convert $s^t$ into $s^t_c$ where $c = 1$
10:         Regroup $s^t_c$ based on $V$ and calculate $|y^t|^v$
11:         Calculate $|z = 1|^v$ for each bin
12:         $P(1|s)^v = \frac{|z=1|^v}{|y^t|^v}$
13:         if $\big(\overline{P(c|s^t_{1i})} \leq \epsilon_1 \wedge \overline{P(c|s^t_{2i})} \geq \epsilon_2\big) \vee \big(A^t > \overline{A}\big)$ then
14:             $t \in RC$
15:     if $\overline{A} > \overline{A}_{max}$ then
16:         Save the current $\epsilon_1$ and $\epsilon_2$
17: while $\lambda_1$ and $\lambda_2$ are not optimal do
18:     Define the new values of $\lambda_1$ and $\lambda_2$
19:     Evaluate $A_{DRV}$
20: for $x \in X_{test}$ do
21:     for $t \in RC$ do
22:         Determine $RC_f$ by the help of the equation 2.
23:     $MajorityVote(RC_f)$
```

4 Experimentation Results and Discussion

To evaluate our algorithm, a series of experiments were performed on eight different datasets. Next section discusses the dataset used, the protocol, and then results are exposed.

4.1 Data Description and Protocol

Table 1. Dataset information.

ID	Dataset	X	Att.		IR
			Num.	Cat.	
1	BCWD	286	0	9	2.365
2	Vertebral	310	6	0	2.1
3	Ionosphere	351	33	0	1.786
4	Musk	476	166	0	1.3
5	Diabetes	768	8	0	1.866
6	Spambase	4601	57	0	1.538
7	Phising	11055	0	30	1.257
8	EES	14980	14	0	1.228

Table 1 provides the information of the dataset that were used in this experiments. Eight datasets from UCI repository [3] were used: Breast Cancer Wisconsin Diagnostic (BCWD), Vertebral Column (Vertebral), Ionosphere, Musk (version 1), Indians Diabetes (Diabetes), Spambase, Phishing Websites, and EEG Eye State (EES). These data are selected in order to study the behavior of our algorithm to handle from BCWD (286 instances and 10 attributes) to EES data (14980 instances and 15 attributes). The attributes vary between numeric (integer and real) and categorical. Since our focus is to study the reliability aspect of various expertise of base predictors, we avoid to use imbalanced dataset so that the performances of the algorithms are not distracted by these conditions. Class imbalance problems can be measured by the imbalance ratio (IR), defined as the ratio of the number of instances in the majority class to the number of examples in the minority class [1]. Balanced data are indicated in Table 1 by the IR score that is close to the value 1. The experiments were conducted by train-test evaluation and the data were split into 67% of training set and 33% of testing set.

Five base classifiers were applied based on different knowledge representations to obtain diversity among the models combination. We used Weka[1] library to build the models of C4.5 (Decision Tree), Naive Bayes (Bayesian), JRip (Rule), Sequential minimal optimization (Function), and k-nearest neighbors (Lazy). Then, we evaluated our proposed algorithm with MV, WMV [9], Stacking, and Multi-Scheme (MS) by accuracy score. The parameters of all algorithms were not changed and we considered the default setting of Weka. Ensemble algorithms were tested in a condition where the base classifiers do not contain a spammer as the first experiment. Then, 25 spammers were added to the base input as second attempt. This second scenario where random predictors are higher than the original classifiers is important to learn the reliability aspect of combination methods [7]. Both experiments were conducted in Java. We set the precision of the threshold p to 0.1 with the interval of the bin equals to 0.1.

[1] http://www.cs.waikato.ac.nz/ml/weka/

4.2 Results and Discussion

Table 2. Accuracy comparison between base classifiers and ensemble methods.

ID Data	C4.5	NB	JRip	SMO	kNN	MV	WMV	DRV	Stacking	MS
1	**0.663**	0.642	0.653	**0.663**	**0.663**	0.653	**0.674**	0.663	0.642	0.642
2	0.786	0.748	0.786	0.796	**0.806**	0.757	0.757	**0.777**	**0.777**	0.748
3	0.897	**0.906**	0.889	**0.906**	0.880	**0.923**	**0.923**	**0.923**	0.889	0.880
4	0.823	0.816	0.709	0.816	**0.835**	0.835	0.835	**0.842**	0.816	0.816
5	0.711	0.719	0.703	0.699	**0.727**	**0.797**	0.770	0.785	0.695	0.789
6	0.907	0.898	0.916	**0.937**	0.926	0.942	**0.943**	**0.943**	0.911	0.937
7	0.943	0.927	0.941	0.940	**0.966**	**0.966**	0.957	**0.966**	0.956	**0.966**
8	0.779	0.671	0.713	0.727	**0.796**	**0.796**	0.766	**0.796**	0.770	**0.796**
mean	0.814	0.791	0.789	0.811	**0.825**	0.834	0.828	**0.837**	0.807	0.822
std dev.	**0.098**	0.112	0.112	0.109	0.100	0.106	**0.103**	**0.103**	0.108	0.105

Table 2 shows the accuracy comparison between base classifiers and ensemble methods. The ID column represents the sequence number of dataset according to the Table 1. The best algorithm is defined by an algorithm that has the highest score of accuracy and the smallest value of standard deviation. kNN shows the best result than the other learners on the side of base classifier. In another way, C4.5 provides the smallest standard deviation among the others. Four out of five ensemble methods exceed the average scores of all single classifiers. This scenario confirms the benefit of ensemble methods to give a better accuracy score than relying on a single classifier. Three voting based algorithms (MV, WMV, DRV) show a superior average results compared to the results of Stacking, and MS. It is normal to see that voting based methods have good results since the base classifiers scores are quite good. WMV and DRV have the same deviation score even though the accuracy score of each dataset is different. If we consider the accuracy score of ensemble methods individually, DRV provides the highest accuracy for six dataset.

Table 3. Accuracy comparison of ensemble methods after 25 spammers were added.

ID Data	MV	WMV	DRV	Stacking	MS
1	0.6	0.653	**0.663**	0.642	0.642
2	0.657	0.755	**0.777**	0.748	**0.777**
3	0.769	0.376	**0.88**	0.863	**0.88**
4	0.747	0.589	0.829	**0.854**	0.816
5	0.664	0.641	**0.734**	0.664	0.711
6	0.799	0.605	0.924	0.918	**0.937**
7	0.815	0.46	**0.966**	0.955	**0.966**
8	0.673	0.559	**0.796**	0.7	**0.796**
mean	0.716	0.580	**0.821**	0.793	0.816
std dev.	**0.077**	0.118	0.100	0.120	0.110

In contrast to the results of the first experiment, Table 3 provides a significant decreasing values on MV and WMV after 25 spammers were added. DRV gives the best result, followed by MS, Stacking, MV, and WMV respectively. MV tends to give similar accuracy score for eight dataset indicated by the smallest standard deviation score, while the accuracy scores of WMV are the lowest among the others on six data. It means that the decision of MV and WMV are distracted by the presence of the spammers, while DRV is able to select the best combination and to eliminate the weak predictions. The ability of ensemble methods to maintain the accuracy score is illustrated in Figure 3. X-axis shows the sequence of the dataset, while Y-axis shows the absolute accuracy distance between the first and second trials (lower value is better). This score is formulated as $\varDelta A = |A_1 - A_2|$, where A_1 is the accuracy value of the first experiment and A_2 is the accuracy score in the presence of spammers. The distance scores of MV are higher than DRV, Stacking, and MS on all dataset, while WMV shows the highest accuracy instability. This measure allows us to see the affect of random predictors to the popular voting techniques. On the other hand, DRV improves this drawback by considering predictor reliability aspect, indicated by the lower score similar to MS and Stacking.

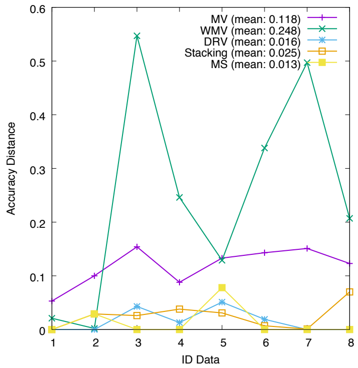
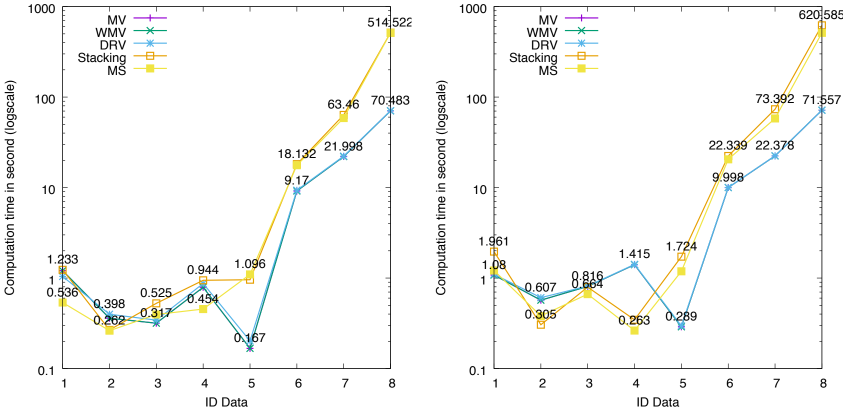


Fig. 3. Accuracy distance before and after 25 spammers were added for eight dataset (smaller value is better).



(a) Five base classifiers (b) Five base classifiers and 25 spammers

Fig. 4. Computation time comparison during training phase (smaller value is better).

Figure 4 illustrates the computation time of five ensemble methods during the training phase. It consists of two conditions where five base classifiers were used as the input (see Fig. 4a) and after the spammers were added (see Fig. 4b). X-axis represents eight data used and Y-axis indicates the number of second needed in a log scale. The values written in the diagram describe the lowest and highest time in each dataset. The performances of MV and WMV were computed from the sum of training time of base classifiers, while the score of DRV was obtained from the MV and the reliability diagram building time (see Eq. 3). Due to the same complexity, MV line is not visible in the figure and is overwritten by the WMV line. According to Figure 4a, all ensemble methods require similar time to train when the number of instances is less than 500. It also shows that the number of instances generally influences the computation time. Although, the performances in BCWD and Diabetes show the opposite results due to their specific characteristics. The superiority of voting based methods compared to Stacking and MS can be seen in Diabetes, Spambase, Phising and EES. Similar results are also presented in Figure 4b. Stacking and MS computed Vertebral, Ionosphere, and Musk faster than the others. In contrary, the deviation between their running time and voting algorithms for the second setup are greater than the first experiments. MV, WMV, and DRV do not have varied results because the spammers do not need significant time to calculate. Based on the comparison of the first and the second figures, the number of base classifiers clearly affects the computation time during the training phase.

5 Conclusion

A diverse group of classifiers are likely to make better decisions comparing to a single learner. However, considering ensemble learning context, each classifier has its own performance. Hence, reliability is a crucial problem when such classifiers have contrasted performances. We propose dynamic reliable voting to solve the problem on how to select the best combination of reliable classifiers and how to handle uncertain labelers, i.e. spammer. The confidence score of prediction is used as main information to produce a reliability diagram of each algorithm and several filters are set to select the best candidates. Five classifiers are chosen as the base models and the voting combination of their predictions for each datum is changed dynamically according to the past experience of their probability estimates. The result shows that our proposed algorithm provides a reliable performance against the previous approaches on eight datasets before and after the presence of spammers. In future work, we will improve our approach to adapt uncertainty and imbalanced class. We will also enhance our algorithm to handle multi-class and multi-label classification.

References

1. García, V., Sánchez, J., Mollineda, R.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems **25**(1), 13 – 21 (2012), special Issue on New Trends in Data Mining

2. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(1), 66–75 (Jan 1994)
3. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
4. Murphy, A.H., Winkler, R.L.: Reliability of subjective probability forecasts of precipitation and temperature. Journal of the Royal Statistical Society. Series C (Applied Statistics) **26**(1), 41–47 (1977)
5. Nachouki, G., Quafafou, M.: Mashup web data sources and services based on semantic queries. Information Systems **25**(2), 151 – 173 (2011)
6. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances In Large Margin Classifiers. pp. 61–74. MIT Press (1999)
7. Raharjo, A.B., Quafafou, M.: The combination of decision in crowds when the number of reliable annotator is scarce. In: Advances in Intelligent Data Analysis XVI, pp. 260–271. Springer International Publishing (2017)
8. Rajnarayan, D., Wolpert, D.: Bias-Variance Trade-offs: Novel Applications, pp. 101–110. Springer US, Boston, MA (2010)
9. Raschka, S.: Mlxtend (Apr 2016). https://doi.org/10.5281/zenodo.594432, http://dx.doi.org/10.5281/zenodo.594432
10. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. J. Mach. Learn. Res. **13**, 491–518 (Feb 2012)
11. Rokach, L.: Ensemble-based classifiers. Artificial Intelligence Review **33**(1), 1–39 (2010)
12. Selwa Elfirdoussi, Z.J., Quafafou, M.: Ranking web services using web service popularity score. International Journal of Information Technology and Web Engineering **9**(2), 78 – 89 (2014)
13. Valdovinos, R.M., Sánchez, J.S.: Combining multiple classifiers with dynamic weighted voting. In: Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems. pp. 510–516. HAIS '09, Springer-Verlag, Berlin, Heidelberg (2009)
14. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Chapter 4 - algorithms: The basic methods. In: Witten, I.H., Frank, E., Hall, M.A., Pal, C.J. (eds.) Data Mining (Fourth Edition), pp. 91 – 160. Morgan Kaufmann, fourth edition edn. (2017)
15. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 694–699. KDD '02, ACM, New York, NY, USA (2002)
16. Zhang, Y., Zhang, H., Cai, J., Yang, B.: A weighted voting classifier based on differential evolution. Abstract and Applied Analysis **2014** (May 2014)