Impacts of Unbalanced Test Data on the Evaluation of Classification Methods

Manh Hung Nguyen[1,2]
[1]Posts and Telecommunications Institute of Technology (PTIT)
[2]UMI UMMISCO 209 (IRD/UPMC), Hanoi, Vietnam

***Abstract*—The performance of a classifier in a supervised machine learning problem is popularly evaluated by using the *accuracy*, *precision*, *recall*, and *F1-score*. These parameters could evaluate very well classifiers in the case that the number of positive label sample and the number of negative label sample in the testing set are balanced or nearly balanced. However, these parameters may miss-evaluate the classifiers in some case where the positive and negative samples in the testing set is unbalanced. This paper proposes some update in these parameters by taking into account the *unbalanced factor* which represents the unbalance ratio of positive and negative samples in the testing set. The new updated parameters are then experimentally evaluated to compare to the traditional parameters.**

***Keywords*—Supervised machine learning evaluation; accuracy; f1 score; unbalanced factor**

I. Introduction

The problem of classification (texts, images, voice...) is already popular in the machine learning community. One of popular methods is supervised machine learning. In which, there are two main phases. First, *training phase*, a set of samples which are already classified with a label, called *training set*, will be used to extract some common features of samples of the same label. This work is done by a classifier. Second, at the *testing phase*, if there is a new sample s, the assignment of a label to the sample s is decided by the classifier trained in the *training phase*.

The performance of the classifier is popularly evaluated by using the *accuracy*, *precision*, *recall*, and *F1-score* parameter which are calculated based on the definition of Salton et al. [7]:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} * 100\% \quad (1)$$

$$Precision = \frac{TP}{TP + FP} * 100\% \quad (2)$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad (3)$$

$$F_1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

where: TP is the number of true positive; FP is the number of false positive; FN is the number of false negative; TN is the number of true negative.

These parameters could evaluate very well classifiers in the case that the number of positive label sample and the number of negative label sample in the testing set are balanced or nearly balanced.

However, these parameters may miss-evaluate the classifiers in some case where the positive and negative samples in the testing set is unbalanced. For instance, let's consider in a case of positive major of testing set in which, there are 90% of samples are positive label and 10% are negative label. There is a very simple classifier which always returns TRUE for any testing sample. In that case, we have:

- $TP = 0.9x$
- $TN = 0$
- $FP = 0.1x$
- $FN = 0$
- $Accuracy = \frac{0.9x + 0}{0.9x + 0.1x + 0 + 0} * 100\% = 90.00\%$
- $Precision = \frac{0.9x}{0.9x + 0.1x} * 100\% = 90.00\%$
- $Recall = \frac{0.9x}{0.9x + 0.1x} * 100\% = 100\%$
- $F_1 - score = 2 * \frac{90 * 100}{90 + 100} = 94.73\%$

where x is the number of sample in the testing set.

With the value of accuracy and F1-score is about 90.00% and 94.75%, respectively, any evaluator could conclude that this is a good classifier. Meanwhile the classifier is very simple and idiot one: it always returns true for any sample. Intuitively, these parameters are lost its objective in this case.

In order to avoid the miss-evaluated in the case of unbalanced testing data, this paper proposes some update in these parameters by taking into account the *unbalanced factor* which represents the unbalance ratio of positive and negative samples in the testing set. The new updated parameters are then experimentally evaluated to compare to the traditional parameters. The paper is organised as follows: Section II presents our proposal of unbalanced factor in the output parameters. Section III presents our experiments to evaluate the proposed update in output parameters. Finally, Section IV is a conclusion.

II. Proposal

We make used the basic concepts based on the definition of Salton et al. [7]:

- *Number of true positive (TP)*: This is the number of samples which are assigned to the considered label.

And in the results, it is also assigned to the same label.

- *Number of false positive (FP)*: This is the number of samples which are NOT assigned to the considered label. But in the results, it is assigned to the label.
- *Number of false negative (FN)*: This is the number of samples which are assigned to the considered label. But in the results, it is NOT assigned to the label.
- *Number of true negative (TN)*: This is the number of samples which are NOT assigned to the considered label. And in the results, it is NOT assigned to the label.

We take into account the *unbalanced factor* which is defined as the ratio between the number of positive sample and that of negative sample in the testing set:

$$\alpha = \frac{\text{number of positive sample in the testing set}}{\text{number of negative sample in the testing set}} \quad (5)$$

This *unbalanced factor* of testing set is then applied in the output parameters by updating the concept of *accuracy*, *precision*, *recall*, and *F-score* as follows:

$$Accuracy = \frac{TP + \alpha * TN}{TP + \alpha * FP + FN + \alpha * TN} * 100\% \quad (6)$$

$$Precision = \frac{TP}{TP + \alpha * FP} * 100\% \quad (7)$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad (8)$$

$$F_1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (9)$$

Intuitively, these updates could replace the *traditional output parameters* in the case the *unbalanced factor* equals to 1. It means that the testing set is balanced or nearly balanced.

Let's return to the paradox example in Section I with a very simple classifier which always returns TRUE for any testing sample, in the case of positive major of testing set in which, there are 90% of samples are positive label and 10% are negative label. If the *unbalanced factor* is taken into account, we will have:

- $TP = 0.9x$
- $TN = 0$
- $FP = 0.1x$
- $FN = 0$
- The unbalanced factor $\alpha = \frac{0.9x}{0.1x} = 9$
- $Accuracy = \frac{0.9x + 9 * 0}{0.9x + 9 * 0.1x + 0 + 9 * 0} * 100\% = 50.00\%$
- $Precision = \frac{0.9x}{0.9x + 9 * 0.1x} * 100\% = 50.00\%$
- $Recall = \frac{0.9x}{0.9x + 0.1x} * 100\% = 100\%$
- $F_1 - score = 2 * \frac{50 * 100}{50 + 100} = 66.67\%$

where x is the number of sample in the testing set.

With the value of accuracy and F1-score is about 50.00% and 66.67%, respectively, any evaluator could conclude that this is a below-average classifier. This is suitable to the classifier which is very simple and idiot one: it always returns true for any sample. Intuitively, these new updated parameters could help us to avoid the case of miss-evaluate the simple classifier in an unbalanced testing set.

III. Evaluation

This section presents an experiment to evaluate the proposed output parameters in the balance and unbalanced testing set.

A. Dataset

This experiment evaluates the proposed model on the dataset of 20 Newsgroups [4]. This dataset contains about 20000 texts, divided into 20 subjects. The longest text has more than 20000 words. The shortest text has about 75 words. The average length of text in this dataset is about 370 words. This dataset is widely used in machine learning and information retrieval domain, in the problem of text classification. The distribution of texts by 20 class labels is presented in Table I.

TABLE I. Distribution of labeled data in the 20 Newsgroups data set

Topics	Number of text
alt.atheism	779
comp.graphics	973
comp.os.ms-windows.misc	985
comp.sys.ibm.pc.hardware	982
comp.sys.mac.hardware	961
comp.windows.x	980
misc.forsale	972
rec.autos	990
rec.motorcycles	994
rec.sport.baseball	994
rec.sport.hockey	999
sci.crypt	991
sci.electronics	981
sci.med	990
sci.space	987
soc.religion.christian	997
talk.politics.guns	910
talk.politics.mideast	940
talk.politics.misc	775
talk.religion.misc	628

B. Scenario

The main scenario of this experiment is defined as follows:

- Using the same training set.
- Using the same classifier. In this experiment, we use the classifier of Multinomial Naive Bayes (MNB) [3]. This algorithm improves the Naive Bayes model with the Multinomial Naive Bayes (MNB) algorithm. It had already proved its good performance in texts

classification as presented in several recent works [5], [6].

- Testing with different sets: balanced testing set, and unbalanced testing set (YES major, and NO major).
- This scenario is repeated in ten times, and then comparing the output parameters in the case with/without unbalanced factor.

1) Building of training set: The training set is built for each label, based on the one-vs-all method [1], as following scenario:

- For each label, select randomly 500 texts whose label is the considered label, and 500 other texts whose label is different from that label.
- Divide this set into ten subsets (for running of ten times): each subset has about 100 texts, in which, 50 texts have the considered label, 50 remain texts have other label.
- For each text in each training subset, remove all stopwords.
- Split the remain character sequence into 1-gram, 2-grams, and 3-grams. The combination of three grams from 1-gram to 3-grams is proved that is the best case for the dataset of 20Newsgroups in the work of Nguyen [5]. That is the reason we use this combination in the experiment.
- Transform it into a vector of TF-IDF [7] value.
- Training with Multinomial Naive Bayes (MNB) [3] classifier[1]

2) Building of testing set: The three testing sets are also built for each label as following scenario:

- Unbalanced testing set with ratio of 20:80 (NO major - called 20:80 testing set):
  - Select randomly 200 texts whose label is that label, and 800 other texts whose label is different from that label.
  - Divide this set into ten subsets (for running of ten times): each subset has about 100 texts, in which, 20 texts have the considered label, 80 remain texts have other label.
- Balanced testing set with ratio of 50:50 (YES/NO balance - called 50:50 testing set):
  - Select randomly 500 texts whose label is that label, and 500 other texts whose label is different from that label.
  - Divide this set into ten subsets (for running of ten times): each subset has about 100 texts, in which, 50 texts have the considered label, 50 remain texts have other label.
- Unbalanced testing set with ratio of 80:20 (YES major - called 80:20 testing set):
  - Select randomly 800 texts whose label is that label, and 200 other texts whose label is different from that label.
  - Divide this set into ten subsets (for running of ten times): each subset has about 100 texts, in which, 80 texts have the considered label, 20 remain texts have other label.
- For each text in each testing subset, remove all stopwords.
- Split the remain character sequence into 1-gram, 2-grams, and 3-grams.
- Transform it into a vector of TF-IDF value.
- Testing with Multinomial Naive Bayes (MNB) classifier.

C. Output Parameters

We consider the output parameters in two cases: without *unbalanced factor* (classical), and with *unbalanced factor* (new proposed).

1) Output parameters without unbalanced factor: In this case, we use the *traditional output parameters* of *Accuracy*, and *F1-score* as the definition of Salton et al. [7] (formula 1 and 4).

2) Output parameters with unbalanced factor: In this case, we take into account the *balance factor* - α of the *testing set*. Therefore, we use the output parameters defined in Section II: accuracy (formula 6), and F1-score (formula 9).

D. Results

The results from the case using output parameters without/with *unbalanced factor* are presented in the Tables II, and III, respectively. These results indicate that the variation of *accuracy* and *F1-score* in the case without *unbalanced factor* is much higher than that in the case with unbalanced factor. For instance, in the case of label *comp.graphics* (the 2nd row in the Tables II and III): The *accuracy* varies from 83.83% to 89.58% and 95.35% in the testing set of 20:80, 50:50, and 80:20 respectively if the *unbalanced factor* is not taken into account. Meanwhile, if the *unbalanced factor* is taken into account, the *accuracy* becomes more stable with value of 90.01%, 89.58%, and 90.30% in the testing set of 20:80, 50:50, and 80:20 respectively.

The same to the value of *F1-score*: It varies from 68.26% to 90.33% and 97.16% in the testing set of 20:80, 50:50, and 80:20 respectively if the *unbalanced factor* is not taken into account. Meanwhile, if the *unbalanced factor* is taken into account, the *F1-score* becomes more stable with value of 90.88%, 90.33%, and 91.08% in the testing set of 20:80, 50:50, and 80:20 respectively.

This principle is appear in almost topics of the considered dataset. Consequently, the average value of *accuracy* and *F1-score* overall 20 topics in the case with *unbalanced factor* are more stable than that in the case without *unbalanced factor* (the last row in the Tables II and III): At the level of *accuracy*, its value varies from 88.35% to 93.07% and 96.08% in the case without *unbalanced factor*. Meanwhile, in

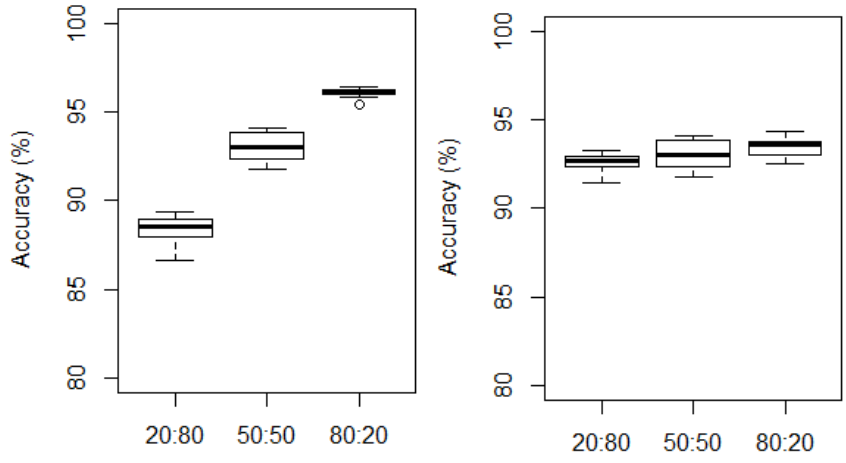
[1]These classifiers are called from API of Weka open source library [2] for Java.

TABLE II. COMPARISON OF ACCURACY AND F1-SCORE (%) WITHOUT THE *unbalanced factor* ON THREE TESTING SETS

Topics	**Accuracy**			**F1-score**		
	20:80 (α=0.25)	50:50 (α=1)	80:20 (α=4)	20:80 (α=0.25)	50:50 (α=1)	80:20 (α=4)
alt.atheism	91.83	95.38	98.08	81.15	95.53	98.82
comp.graphics	83.83	89.58	95.35	68.26	90.33	97.16
comp.os.ms-windows.misc	94.17	90.03	86.87	84.95	89.24	91.12
comp.sys.ibm.pc.hardware	76.78	87.06	94.24	59.88	88.47	96.53
comp.sys.mac.hardware	82.87	90.22	95.35	67.02	90.94	97.13
comp.windows.x	89.04	92.74	95.76	75.96	92.97	97.35
misc.forsale	85.30	90.81	95.35	70.46	91.39	97.13
rec.autos	90.26	93.27	96.57	78.34	93.60	97.88
rec.motorcycles	90.78	94.96	97.17	79.07	95.18	98.25
rec.sport.baseball	92.35	96.41	97.47	82.15	96.54	98.44
rec.sport.hockey	94.96	97.66	98.08	87.42	97.72	98.80
sci.crypt	87.30	94.50	97.37	73.60	94.77	98.39
sci.electronics	85.91	91.16	95.15	71.47	91.65	97.02
sci.med	88.43	94.07	97.27	75.81	94.31	98.19
sci.space	93.57	95.61	97.07	84.65	95.73	98.67
soc.religion.christian	96.00	97.79	98.28	89.81	97.85	98.94
talk.politics.guns	88.17	93.84	96.87	74.86	94.09	98.07
talk.politics.mideast	94.43	96.97	98.69	86.26	97.04	99.19
talk.politics.misc	73.48	87.00	94.75	57.38	88.47	96.84
talk.religion.misc	87.57	92.32	95.76	73.73	92.76	97.38
Average	**88.35**	**93.07**	**96.08**	**76.11**	**93.43**	**97.55**

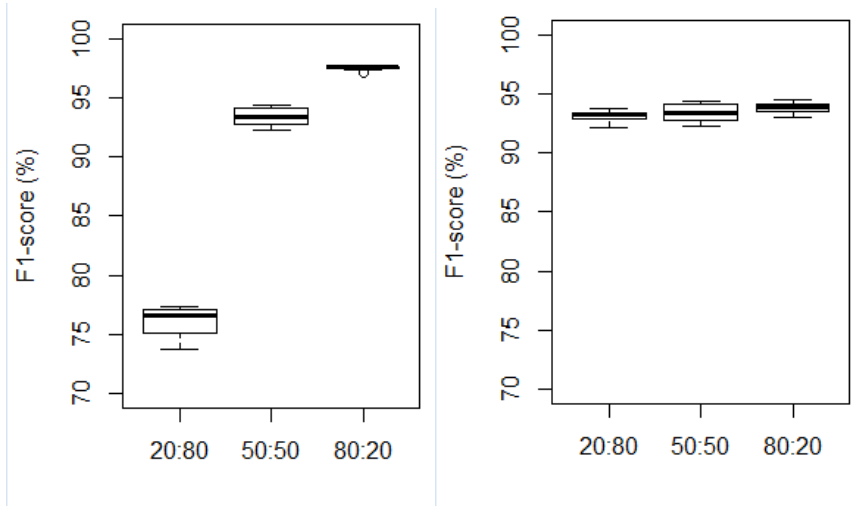
TABLE III. COMPARISON OF ACCURACY AND F1-SCORE (%) WITH THE *unbalanced factor* ON THREE TESTING SETS

Topics	**Accuracy**			**F1-score**		
	20:80 (α=0.25)	50:50 (α=1)	80:20 (α=4)	20:80 (α=0.25)	50:50 (α=1)	80:20 (α=4)
alt.atheism	94.66	95.38	96.40	94.89	95.53	96.52
comp.graphics	90.01	89.58	90.30	90.88	90.33	91.08
comp.os.ms-windows.misc	93.12	90.03	90.27	93.00	89.24	89.60
comp.sys.ibm.pc.hardware	85.55	87.06	87.01	87.28	88.47	88.54
comp.sys.mac.hardware	89.43	90.22	91.71	90.41	90.94	92.20
comp.windows.x	92.78	92.74	94.16	93.17	92.97	94.35
misc.forsale	90.91	90.81	92.31	91.65	91.39	92.76
rec.autos	94.11	93.27	94.06	94.45	93.60	94.39
rec.motorcycles	94.22	94.96	94.84	94.52	95.18	95.08
rec.sport.baseball	95.17	96.41	95.63	95.38	96.54	95.82
rec.sport.hockey	96.75	97.66	97.81	96.84	97.72	97.86
sci.crypt	92.12	94.50	94.36	92.69	94.77	94.73
sci.electronics	91.28	91.16	91.18	91.97	91.65	91.79
sci.med	93.00	94.07	94.70	93.52	94.31	94.96
sci.space	95.71	95.61	95.58	95.86	95.73	95.81
soc.religion.christian	97.58	97.79	97.33	97.64	97.85	97.43
talk.politics.guns	92.84	93.84	94.25	93.34	94.09	94.56
talk.politics.mideast	96.43	96.97	97.58	96.54	97.04	97.70
talk.politics.misc	83.95	87.00	87.32	86.26	88.47	88.78
talk.religion.misc	92.47	92.32	92.56	93.01	92.76	92.99
Average	**92.60**	**93.07**	**93.47**	**93.17**	**93.43**	**93.85**



(a) without unbalanced factor (b) with unbalanced factor

Fig. 1. Variation of Accuracy in three testing sets in the case without and with *unbalanced factor*.



(a) without unbalanced factor (b) with unbalanced factor

Fig. 2. Variation of F1-score in three testing sets in the case without and with *unbalanced factor*.

the case with *unbalanced factor*, its value has a small change from the 92.60% to 93.07% and 93.47% in the testing set of 20:80, 50:50, and 80:20 respectively. At the level of *F1-score*, its value varies from 76.11% to 93.43% and 97.55% in the case without *unbalanced factor*. Meanwhile, in the case with *unbalanced factor*, its value has a small change from the 93.17% to 93.43% and 93.85% in the testing set of 20:80, 50:50, and 80:20 respectively.

In order to see the difference in detail from the two considered cases, we compared the results from ten times of testing on each output parameters. At the level of *accuracy* (Fig. 1), its value in the case without *unbalanced factor* is significantly different from the testing set of 20:80, 50:50, and 80:20 (Fig. 1(a)). Meanwhile, there is no significant difference from its value in the case with *unbalanced factor* (Fig. 1(b)): this value is stably about 93%. The same results at the level of *F1-score* (Fig. 2), its value in the case without *unbalanced factor* is significantly different from the testing set of 20:80, 50:50, and 80:20 (Fig. 2(a)). Meanwhile, there is no significant difference from its value in the case with *unbalanced factor* (Fig. 2(b)): this value is stably within 93-94%.

In summary, the experiment results indicate that the *unbalanced factor* could bring the value of *accuracy* and *F-score* of a classification method more stable. In other words, it could make the value of *accuracy* and *F-score* of a classification method more independent from the unbalanced ratio of label in the testing set.

IV. Conclusion

This paper proposed some update in the output parameters in evaluation of supervised machine learning methods (accuracy, precision, recall, F1-score) by taking into account the *unbalanced factor* which represents the unbalance ratio of

positive and negative samples in the testing set. The new updated parameters are then experimentally evaluated to compare to the traditional parameters. The experiment results indicate that the new updated parameters could evaluate the classifier with a stable value in spite of the change of unbalanced ratio between the positive and negative samples in the testing set.

REFERENCES

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations Newsletter*, 11(1):10–18, November 2009.

[3] Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence*, AI'04, pages 488–499, Berlin, Heidelberg, 2004. Springer-Verlag.

[4] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[5] Manh Hung Nguyen. On the distinction of subjectivity and objectivity of emotions in texts. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(9):584–589, 2018.

[6] R.G. Rossi, R. M. Marcacini, and S. O. Rezende. Benchmarking text collections for classification and clustering tasks. Technical Report 395, Institute of Mathematics and Computer Sciences - University of Sao Paulo, 2013.

[7] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.