A New Evaluation Measure for Learning from Imbalanced Data

Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme, *Member, IEEE*

***Abstract*— Recently, researchers have shown that the Area Under the ROC Curve (AUC) has a serious deficiency since it implicitly uses different misclassification cost distributions for different classifiers. Thus, using the AUC can be compared to using different metrics to evaluate different classifiers [1]. To overcome this incoherence, the H measure was proposed, which uses a symmetric Beta distribution to replace the implicit cost weight distribution in the AUC. When learning from imbalanced data, misclassifying a minority class example is much more serious than misclassifying a majority class example. To take different misclassification costs into account, we propose using an *asymmetric* Beta distribution (B42) instead of a symmetric one. Experimental results on 36 imbalanced data sets using SVMs and logistic regression show that B42 is a good choice for evaluating on imbalanced data sets because it puts more weight on the minority class. We also show that balanced random undersampling does not work for large and highly imbalanced data sets, although it has been reported to be effective for small data sets.**

I. Introduction

Class imbalance is a phenomenon in which the class distribution[1] is far from the uniform distribution. It appears in many machine learning applications such as fraud detection, intrusion detection, and so on [2], [3]. Most classifiers are designed to maximize the accuracy of their models. Thus, when learning from imbalanced data, they are usually overwhelmed by the majority class examples. This is the main cause for the performance degradation of such classifiers, and is also considered as one of ten challenging problems in data mining research [4]. For example, in fraud credit card detection, suppose that the data set has 999 legitimate transactions (majority class) and only 1 fraudulent transaction (minority class – the one we would like to detect). To maximize the accuracy, in this case, classifiers optimized for accuracy will classify all transactions as belonging to the majority class to get 99.9% accuracy. However, this result has no meaning because the fraudulent transaction is misclassified.

Obviously, to evaluate the classifiers in this case, the accuracy metric becomes useless, and the area under the ROC curve (AUC) is commonly used instead [5], [6]. The AUC has been widely used to evaluate the performance of classifiers. Recently, [1] has shown that using the AUC is equivalent to averaging the misclassification loss over a cost ratio distribution which depends on the score distributions. Since the score distributions depend on the classifier itself, employing the AUC as an evaluation measure actually means measuring different classifiers using different metrics. To overcome this incoherence, the "H measure" was proposed, which uses a symmetric Beta distribution to replace the implicit cost weight distribution in the AUC. When learning from imbalanced data, misclassifying a minority class example (e.g, a fraud credit card transaction) is much more serious than misclassifying a majority example. Thus, we propose using an asymmetric Beta distribution such as $beta(x; 4, 2)$ (called **B42**) instead of the symmetric one as in the H measure.

Moreover, many papers have been published about the class imbalance problem, but there is still little insight on how skew class distributions affect the classifiers when learning from *large and imbalanced* data sets. Furthermore, as investigated in [3], there are two open problems for the future research in this area: The need for a *standardized evaluation* protocol and the need for *uniform benchmarks* as well as *large data sets* [7]. The contributions of this work are *(1)* to propose an evaluation metric for learning from imbalanced data, *(2)* to introduce large benchmark data sets for systematic studies on imbalanced data, and *(3)* to investigate the influence of class imbalance on the behavior of classifiers when learning from large data sets.

The rest of the paper is organized as follows. Section II introduces the H metric followed by the proposed B42 metric; in session III, we summarize some common techniques that are usually used to tackle the class imbalance problem; section IV first presents the evaluation protocol and the data sets. Then, we analyze and compare the results of three metrics (B42, AUC, and H) followed by analyzing the behaviors of classifiers when learning from large and imbalanced data; and finally, section V concludes the article.

II. New Evaluation Measures

A. The H Measure – A Replacement for the AUC

To overcome the incoherence of the AUC, the "H measure" was proposed, which is determined by

$$H = 1 - \frac{\int Q(T(c); b, c) u_{\alpha,\beta}(c) dc}{\pi_0 \int_0^{\pi_1} c u_{\alpha,\beta}(c) dc + \pi_1 \int_{\pi_1}^{1} (1-c) u_{\alpha,\beta}(c) dc}. \quad (1)$$

where π_0 and π_1 are prior probabilities; c_0 and c_1 are the misclassification costs for class 0 (majority) and class 1 (minority); $b = c_0 + c_1$ and $c = c_1/(c_0 + c_1)$; $f_0(s)$ and $f_1(s)$ are the probability density functions; and $F_0(s)$ and $F_1(s)$ are the cumulative distribution functions for class 0

Nguyen Thai-Nghe is with the Information Systems and Machine Learning Lab, University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany (phone: +49 5121 883 765; email: nguyen@ismll.de).

Zeno Gantner is with the Information Systems and Machine Learning Lab, University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany (phone: +49 5121 883 856; email: gantner@ismll.de).

Lars Schmidt-Thieme is with the Information Systems and Machine Learning Lab, University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany (phone: +49 5121 883 851; email: schmidt-thieme@ismll.de)

[1]In this paper, we consider the problem of binary classification.

and class 1, respectively.

$$Q(t;b,c) \triangleq \{c\pi_1(1-F_1(t)) + (1-c)\pi_0 F_0(t)\}b$$

is the loss for an arbitrary choice of threshold t and

$$u_{\alpha,\beta} = beta(c;\alpha,\beta) = \frac{c^{\alpha-1}(1-c)^{\beta-1}}{B(1;\alpha,\beta)}$$

is a symmetric Beta distribution. Please refer to [1], [8] for details.

B. B42 – A New Evaluation Measure for Learning from Imbalanced Data

Beta distributions are a popular model for random variables [9] with values in the interval [0,1]. The Beta function, also known as Euler's Beta integral [9], is defined as

$$B(1;\alpha,\beta) = \int_0^1 c^{\alpha-1}(1-c)^{\beta-1}dc.$$

It can also be defined by using the Gamma function

$$B(1;\alpha,\beta) = \frac{\Gamma(\alpha)\,\Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

A generalization of the Beta function is the incomplete Beta function:

$$B(x;\alpha,\beta) = \int_0^x c^{\alpha-1}(1-c)^{\beta-1}dc.$$

The probability density function of the Beta distribution has its mode at $\frac{\alpha-1}{\alpha+\beta-2}$ and is determined by

$$f(x;\alpha,\beta) = \frac{1}{\mathrm{B}(1;\alpha,\beta)}\,x^{\alpha-1}(1-x)^{\beta-1}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}.$$

As discussed in [1], the alternative cost distribution, which can replace the implicit cost weight distribution in the AUC, needs to be a non-uniform one. Thus, an asymmetric Beta distribution would be a good choice for this replacement.

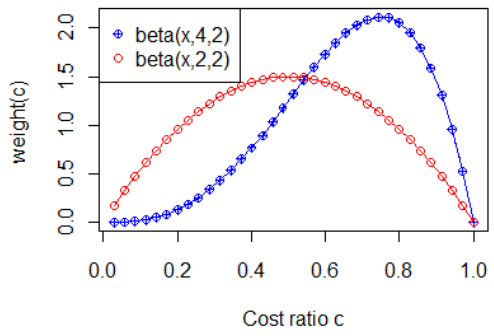

Fig. 1. Symmetric and Asymmetric Beta Distributions

As we can see in Figure 1, for two balanced classes, a symmetric Beta distribution acts as a cost weight distribution, which places most probabilities at 0.5, is used in the H.

However, when learning from imbalanced data sets, misclassifying a minority class example (e.g, in terrorist detection system, misclassifying a terrorist who can carry a bomb on a flight) is much more serious than misclassifying a majority class example (e.g, misclassifying a normal passenger as a terrorist) [10]. Thus, the misclassification cost c_1 (false negative cost) of the minority is much higher than the misclassification cost c_0 (false positive cost) of the majority, therefore, the cost ratio $c = c_1/(c_0+c_1)$ should be higher than 0.5. For the aforementioned reason, we use the *asymmetric* Beta distribution B42 as a cost weight distribution. B42 *places higher weight on minority class examples* and is a unimodal distribution with mode at 0.75.

Please note that one can choose some other values for α (e.g, *beta(x,6,2)*, *beta(x,8,2)*...). In those cases, the absolute values of the metrics can be higher, but the relative values are not significantly different. Thus, we decide to use *beta(x,4,2)*.

III. Dealing with Class Imbalance

To deal with imbalanced data sets, many techniques have been introduced, e.g. undersampling [11], [12]; oversampling [13]; manipulating classifiers internally [14], [15], [16]; cost-sensitive learning [17], [18], [10]; and more [2], [3].

In this work, we have not focused on designing or improving the performance of the classifiers but on a new *evaluation metric* for imbalanced data sets learning. Different from the H measure [1] which uses a symmetric Beta distribution, we propose using an asymmetric one to put more weight on the minority class, thus, it is more appropriate for learning from imbalanced data.

For analyzing the behavior of the classifiers, we use different costs (weights) for different classes, which sets different values of parameter C for different target classes [14]. We will call this the *weighting method* in the rest of the article. In this method, given a data set $\mathcal{D}$ consisting of n examples (x_i, y_i), where $x_i \in \mathcal{X}$ are input features and $y_i \in \{-1,+1\}$ is the target class; n_+ and n_- are number of positive (minority) and negative (majority) examples. A linear SVM for imbalanced data solves the following unconstrained optimization problem:

$$\min_w \frac{1}{2}w^T w + C^+\left(\sum_{\{i|y_i=+1\}}^{n_+}\xi_i\right) + C^-\left(\sum_{\{j|y_j=-1\}}^{n_-}\xi_j\right), \quad (2)$$

where C^+ and C^- are penalty values for minority and majority class examples. For imbalanced data, the separating hyperplane needs to be pushed towards the positive examples, thus, *C^+ will be assigned a greater value than C^-*.

Akbani [15] combined SMOTE – an oversampling method [2] – with the weighting method to cope with imbalanced data. In this study, working on large data sets, oversampling needs lots of memory and training time, so we only use the weighting approach.

TABLE I

COMPARISON OF ℓ_2-SVM (BASE) AND ℓ_2-LR USING THREE METRICS: B42, AUC, AND H

			B42			AUC			H		
Data set	%Minority	Size	ℓ_2-SVM	ℓ_2-LR		ℓ_2-SVM	ℓ_2-LR		ℓ_2-SVM	ℓ_2-LR	
r2l (*)	0.02	743.0 MB	0.413±.371	0.519±.356		0.963±.082	0.980±.044		0.345±.290	0.444±.278	
nf-005p (***)	0.05	2.6 GB	0.006±.002	0.005±.003		0.523±.033	0.617±.038	○	0.002±.001	0.003±.002	
nf-05p (**)	0.50	2.6 GB	0.005±.001	0.022±.005		0.628±.008	0.767±.013	○	0.002±.001	0.010±.001	
probe (*)	0.83	743.0 MB	0.358±.364	0.543±.283		0.726±.119	0.818±.122		0.324±.270	0.467±.196	
nf-1p(***)	1.00	2.6 GB	0.010±.001	0.039±.002		0.670±.007	0.784±.004	○	0.005±.001	0.019±.001	
appetency (**)	1.78	1.6 GB	0.012±.003	0.026±.007		0.735±.020	0.775±.014	○	0.007±.003	0.013±.005	
ann	2.30	436.0 KB	0.615±.093	0.659±.068		0.929±.025	0.984±.010	○	0.536±.105	0.591±.057	
allhyper	2.70	270.0 KB	0.459±.125	0.328±.105	●	0.862±.084	0.886±.030		0.254±.226	0.227±.116	
w1a	2.97	3.4 MB	0.169±.183	0.214±.106	○	0.810±.108	0.853±.034		0.108±.124	0.160±.133	○
allrep	3.29	275.0 KB	0.441±.064	0.385±.058	●	0.970±.006	0.967±.008		0.343±.072	0.264±.042	●
anneal	4.45	80.0 KB	0.635±.155	0.411±.116	●	0.957±.028	0.911±.042		0.573±.181	0.392±.297	
allbp	4.75	200.0 KB	0.374±.169	0.324±.082		0.886±.135	0.859±.112		0.280±.132	0.207±.081	
hypothyroid	4.77	281.0 KB	0.134±.179	0.343±.139	○	0.834±.103	0.843±.056		0.292±.208	0.266±.143	
nf-5p (***)	5.00	2.6 GB	0.112±.005	0.126±.005		0.766±.036	0.804±.004		0.061±.003	0.068±.003	
sick	6.10	205.0 KB	0.625±.071	0.596±.065	●	0.929±.035	0.941±.023		0.535±.080	0.517±.070	
churn (**)	7.34	1.6 GB	0.011±.003	0.023±.005		0.605±.017	0.648±.018	○	0.005±.002	0.011±.002	
abalone	9.36	259.0 KB	0.206±.054	0.205±.054		0.847±.024	0.845±.024		0.125±.043	0.122±.041	
ijcnn	9.70	7.6 MB	0.313±.158	0.300±.150		0.861±.059	0.858±.058		0.227±.144	0.214±.135	
nf-10p (***)	10.00	2.6 GB	0.194±.008	0.226±.010	○	0.756±.005	0.817±.006	○	0.118±.006	0.137±.007	
nf-20p (***)	20.00	2.6 GB	0.223±.004	0.237±.003		0.752±.003	0.772±.003		0.149±.003	0.157±.003	
hepatitis	20.64	23.0 KB	0.422±.195	0.484±.147		0.645±.368	0.736±.257		0.344±.234	0.417±.341	
transfusion	23.80	24.0 KB	0.060±.077	0.399±.253	○	0.562±.177	0.761±.178		0.132±.142	0.372±.255	
a9a	23.93	3.4 MB	0.266±.033	0.270±.009		0.792±.006	0.794±.005		0.176±.006	0.178±.007	
a2a	24.08	2.3 MB	0.293±.011	0.318±.011	○	0.792±.009	0.793±.005		0.178±.013	0.180±.007	
real-sim	30.75	88.2 MB	0.474±.278	0.768±.200	○	0.812±.218	0.959±.057		0.455±.263	0.735±.231	○
url	33.05	2.2 GB	0.075±.021	0.095±.034		0.546±.025	0.565±.045		0.072±.020	0.086±.032	
cod-rna	33.30	25.4 MB	0.166±.115	0.108±.076		0.586±.226	0.640±.094		0.155±.106	0.079±.056	
pima	34.89	41.0 KB	0.216±.144	0.169±.085		0.587±.197	0.621±.037		0.186±.124	0.158±.077	
diabetes	34.90	68.0 KB	0.396±.052	0.398±.049		0.671±.055	0.695±.052		0.144±.059	0.165±.072	
heartdisease	36.00	22.0 KB	0.024±.034	0.108±.090	○	0.317±.177	0.550±.105	○	0.023±.033	0.092±.068	
breastcancer	37.99	60.0 KB	0.087±.096	0.138±.144		0.404±.158	0.488±.176	○	0.099±.112	0.150±.154	
nf-47p (***)	47.00	2.6 GB	0.006±.001	0.007±.000		0.463±.003	0.462±.002		0.007±.001	0.008±.001	
rcv1	47.54	1.2 GB	0.006±.004	0.086±.022	○	0.533±.015	0.559±.025		0.006±.004	0.063±.017	○
splice	48.30	699.0 KB	0.106±.024	0.111±.027		0.584±.030	0.589±.031		0.084±.020	0.086±.021	
covtype	48.76	70.0 MB	0.087±.101	0.103±.112		0.533±.106	0.543±.108		0.072±.084	0.084±.090	
news20	49.99	136.7 MB	0.099±.074	0.088±.046		0.490±.251	0.486±.241		0.101±.169	0.091±.146	
Average			.225	.256		.703	.749		.181	.202	

(*): KDD Cup 1999 data set; (**): KDD Cup 2009 data set; (***): Netflix data set.
○, ● statistically significant improvement or degradation (level=0.05).

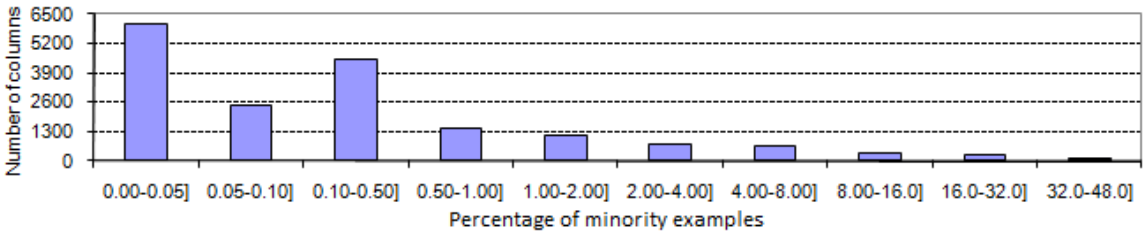

Fig. 2. Distribution of columns and %minority examples on Netflix data set

IV. EMPIRICAL EVALUATION

A. Protocol

We compare two classifiers – ℓ_2-regularized logistic regression (ℓ_2-LR) and ℓ_2-loss SVMs (ℓ_2-SVM) – wrt. the AUC, H, and B42 on 36 data sets using 5-fold cross-validation. To test for significance, we perform paired t-tests with significance level 0.05. We use the LIBLINEAR software [19] with some small modifications to get posterior probability outputs.

For analyzing classifier behavior, we have compared the performance of the classifiers when learning on original data sets with two other methods: random undersampling until two class distributions are balanced (RUS-balance) and using different weights for different classes (weighting) as in equation (2). We have not tried other methods (e.g, oversampling, data cleaning, ...) since these methods need a lot of memory and training time for large data sets. Moreover, we only aim at analyzing the behavior of single classifiers, so we do not take other advanced methods or ensembles, e.g [20], [12], [18], [16], into account. We perform hyperparameter search as described in [10] to determine the best hyperparameters for all methods, e.g the ratio between C^+ and C^-, since our previous results shown that this solution was helpful [10], [21].

B. Data Sets

We have experimented on both small and large data sets collected from the UCI repository[2] and the Netflix Prize[3]. We group them into 3 groups as in Table I. Nominal attributes are converted to binary numeric attributes. For multi-class data sets, many of them (e.g, RCV1, News20, etc.) were already transformed to binary-class data sets as in the LIBSVM data set library[4]. The remaining multi-class datasets are converted to binary-class using one-versus-the-rest. We encoded the class which has the smallest number of examples as the minority (positive) class, and the rest as the majority (negative) one.

The Netflix (nf) data set originally has 100,480,507 ratings from 480,189 customers for 17,770 movies. To create a binary matrix, in which rows represent users/customers and columns represent items/movies, we assign 1 for each observed rating, and 0 otherwise. We then sort the columns based on their class distributions as in Figure 2. To create a data set, we choose one column (movie) to be the target, whereas the other columns represent the input features. This way, we can generate 17,770 different data sets. For example, the data set "nf-05p" means that we choose a target column which has 0.5% minority.

Please note that the last five data sets are not imbalanced. We use them to see how the results are affected when learning from "nearly balanced" to "highly imbalanced" class distributions.

C. B42 versus AUC and H

Table I presents the detailed results of three metrics: B42, AUC, and H. The AUC evaluates ℓ_2-LR outperforming ℓ_2-SVM (at least equal) on 3 groups, while B42 shows that when the imbalance ratio increases, ℓ_2-LR shifts from win (3/9/0) to lose (1/8/3) results, as illustrated in Table II. For example, 1/8/3 means that the ℓ_2-LR wins one time, ties eight times, and loses three times, compared to the ℓ_2-SVM.

Tables III and IV summarize the agreed/disagreed results of B42 vs. AUC and B42 vs. H on 36 data sets when comparing ℓ_2-LR with ℓ_2-SVM (base). The bold number in the diagonal (e.g. 10 and 7) means that B42 evaluates ℓ_2-LR significantly outperforming/degrading ℓ_2-SVM 10 times, but that AUC disagrees on those results, while the reverse is 7 times the case. These agreed/disagreed results could be because the B42 places more weight on the minority examples, thus, it has more statistically significant improvements or degradations compared to the AUC and the H. However, a deeper analysis needs to be done here. The results are presented in the next paragraph.

[2]http://archive.ics.uci.edu/ml/
[3]http://www.netflixprize.com
[4]http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

TABLE II
WIN/TIE/LOSE RESULTS AGGREGATED FROM TABLE I TO 3 GROUPS; ℓ_2-SVM (BASE) VS. ℓ_2-LR

Groups (12 data sets)	%Minority	B42	AUC	H
Group 1 (highly imba.)	0.02 - 5	**1/8/3**	5/7/0	1/10/1
Group 2	5 - 30	4/7/1	2/10/0	0/12/0
Group 3 (nearly bala.)	30 - 49	**3/9/0**	2/10/0	2/10/0

TABLE III
THE B42 DISAGREES WITH THE AUC 17 TIMES OUT OF 36 DATA SETS

	Signif. diff.	Not signif. diff.
Significantly different	2	**7**
Not significantly different	**10**	17

TABLE IV
THE B42 DISAGREES WITH THE H 8 TIMES OUT OF 36 DATA SETS

	Signif. diff.	Not signif. diff.
Significantly different	4	**0**
Not significantly different	**8**	24

Let us analyze more details for the specific data set "nf-05p" in Figure 3, which displays an example of cost weight distribution implicitly used in the AUC (for "nf-05p") and explicitly used in B42 and H.

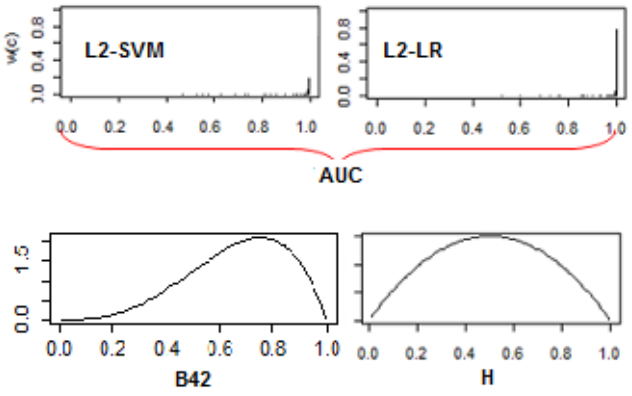

Fig. 3. Cost weight distribution of the AUC (on nf-05p data set), of B42, and of H

Clearly, the AUC places *different cost weight distributions for ℓ_2-LR (higher at 1.0) and ℓ_2-SVM on the same "nf-05p"*

data set. This means that the AUC uses different metrics to evaluate different classifiers [1], while B42 and H use the same distribution for all data sets and classifiers. This is the reason why the result of ℓ_2-LR significantly outperforms ℓ_2-SVM regarding the AUC while it only ties regarding B42. The same situation happens with other data sets e.g, "nf-005p", "nf-1p" and "ann".

Furthermore, Figure 4 shows four typical results of the AUC, the true positive rate, and the B42. We can see that the AUC evaluates the ℓ_2-LR outperforming the ℓ_2-SVM, however, the true positive rate and the B42 show the reversed results.

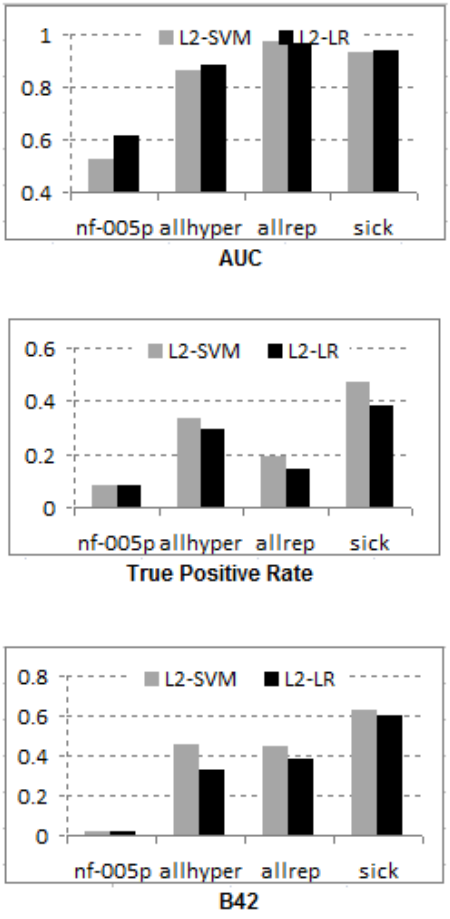

Fig. 4. Typical results of the AUC, the True positive rate, and the B42

The B42 is consistent with the true positive rate while the AUC is not. Thus, if we would like to take the minority class into account then the B42 is a better choice.

In addition, the empirical results also show that B42 is not only suitable for evaluating on imbalanced data but also for evaluating on balanced data sets (in group 3 in Table I, its results are also consistent with other metrics, e.g the H measure).

D. The Influence of Class Imbalance on Large Data Sets

We analyzed on 7 large data sets from Table I (the bold names). The results are reported in Table V. We compared learning on the original data with two other methods: the RUS-balance and the weighting. While the weighting method works fine, the RUS-balance degrades the classifier significantly. This could be because of much information is discarded by undersampling. These results contradict previous studies (e.g, in [22], [23]) which conducted experiments on small data sets. However, more works are needed to be done here.

When the data set is highly imbalanced (e.g, 0.05% minority as in *nf-005p*), the B42 score is low (e.g, 0.005 on original data). The B42 score increases when the imbalance ratio decreases (e.g, to 0.226 at 10% minority). This phenomenon happens not only for the original data but also for the weighting. Thus, this means that the class imbalance affects the classifiers systematically.

To find out why RUS-balance did not work on large data sets, we also looked at the true positive rate and true negative rate before and after dealing with class imbalance. We found that the true positive rate increases while the true negative rate decreases significantly. Typical results are shown in Figure 5.

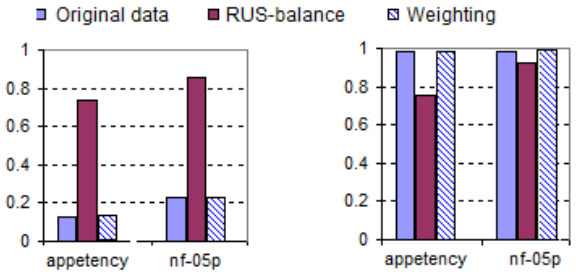

Fig. 5. True positive rate (left) and true negative rate (right)

This phenomenon shows that in the small data sets, the number of false positive is small so we could not see the negative effect on the results, but in case of large data, this number may also be large if we highly focus on minority class examples. Thus, the performance of overall model degrades significantly although the number of false negatives has decreased by dealing with class imbalance.

To this end, one can see that the trade-off between the false negative and the false positive should be taken into account when learning on large and imbalanced data, and the RUS-balance is not a solution on large data sets. We propose treating the undersampling ratio as a hyperparameter, and search for the best one. This method does not significantly improve the classifier performance but at least does not worsen the classifier performance and can reduce the memory consumption.

V. Conclusion

We propose the asymmetric Beta distribution B42 to evaluate classifiers when learning from imbalanced data sets, instead of using the AUC, which has known shortcomings, and the H measure, which fixes the AUC's deficiencies, but is more suitable for balanced class distributions. The

TABLE V

THE INFLUENCE OF CLASS IMBALANCE ON LARGE DATA SETS. RESULTS OF THE B42 USING ℓ_2-LR

Data set	#Examples	#Features	Size	%Minority	Original Data	RUS-balance	Weighting
appetency	87,904	15,000	1.6GB	1.78	0.026± 0.007	0.001±0.001	0.031±0.007
churn	87,904	15,000	1.6GB	7.34	0.023± 0.005	0.001±0.000	0.030±0.005
nf-005p	480,189	17,770	2.6GB	0.05	**0.005**± 0.003	0.001±0.000	0.002±0.003
nf-05p	480,189	17,770	2.6GB	0.50	0.022± 0.003	0.001±0.000 •	0.005±0.001
nf-1p	480,189	17,770	2.6GB	1.00	0.039± 0.002	0.001±0.000 •	0.081±0.005 ○
nf-5p	480,189	17,770	2.6GB	5.00	0.126± 0.005	0.001±0.000 •	0.166±0.005 ○
nf-10p	480,189	17,770	2.6GB	10.0	**0.226**± 0.010	0.003±0.001 •	0.329±0.004 ○

○, • *statistically significant improvement or degradation*

experiments show that the results for the AUC and B42 are reversed for highly imbalanced data and the B42 can take the minority class into account when evaluating. We also analyze how the class imbalance affects the behavior of classifiers and find out why the RUS-balance fails when learning from large data sets. In the future, we will study how to directly optimize the B42 and H measures.

ACKNOWLEDGMENTS

The first author was funded by the "Teaching and Research Innovation Grant" Project of Cantho University, Vietnam.

REFERENCES

[1] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve," *Machine Learning*, vol. 77, no. 1, pp. 103–123, October 2009.

[2] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.

[3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, September 2009.

[4] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology and Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

[5] J. Hanley and B. Mcneil, "The meaning and use of the area under receiver operating characteristics (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[6] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 7, no. 30, pp. 1145–1159, 1997.

[7] A. Jamain and D. J. Hand, "Where are the large and difficult datasets?" *Advances in Data Analysis and Classification*, vol. 3, no. 1, pp. 25–38, June 2009.

[8] D. J. Hand, "Classifier technology and the illusion of progress," *Statistical Science*, vol. 21, no. 1, pp. 1–14, Jun 2006.

[9] M. H. Degroot and M. J. Schervish, *Probability and Statistics, 3rd Edition*. Addison-Wesley, 2002.

[10] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *Proceeding of IEEE International Joint Conference on Neural Networks (IJCNN 2010)*. IEEE Xplore, 2010, pp. 1–8.

[11] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[12] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 2, pp. 539–550, April 2009.

[13] N. V. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligent Research*, vol. 16, pp. 321–357, 2002.

[14] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI 1999)*, 1999, pp. 55–60.

[15] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, 2004, pp. 39–50.

[16] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, "A robust decision tree algorithm for imbalanced data sets," in *SIAM International Conference on Data Mining (SDM'10)*, 2010, pp. 766–777.

[17] C. Elkan, "The foundations of cost-senstive learning," in *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI 2001)*, 2001, pp. 973–978.

[18] X.-Y. Liu and Z.-H. Zhou, "Learning with cost intervals," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 403–412.

[19] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, August 2008.

[20] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Statistical Analysis and Data Mining*, vol. 2, no. 5-6, pp. 412–426, 2009.

[21] N. Thai-Nghe, A. Busche, and L. Schmidt-Thieme, "Improving academic performance prediction by dealing with class imbalance," in *Proceedings of the 9th IEEE International Conference on Intelligent Systems Design and Applications (ISDA 2009)*. IEEE Computer Society, 2009, pp. 878–883.

[22] G. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–26, 2004.

[23] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of International Conference on Machine Learning (ICML 2007)*, 2007, pp. 935–942.