Learning Optimal Threshold on Resampling Data to Deal with Class Imbalance

Nguyen Thai-Nghe
Machine Learning Lab
University of Hildesheim
31141 Hildesheim, Germany
nguyen@ismll.de

Thanh-Nghi Do
College of ICT
Can Tho University
Can Tho City, Vietnam
dtnghi@cit.ctu.edu.vn

Lars Schmidt-Thieme
Machine Learning Lab
University of Hildesheim
31141 Hildesheim, Germany
schmidt-thieme@ismll.de

***Abstract*—Class imbalance is one of the challenging problems for machine learning algorithms. When learning from highly imbalanced data, most classifiers are overwhelmed by the majority class examples, thus, their performance usually degrades. Many papers have been introduced to tackle this problem including methods for pre-processing, internal classifier processing, and post-processing – which mainly relies on the posterior probabilities. Bayesian Network (BN) is known as a classifier which can produce good posterior probabilities. In this study, we propose methods to combine resampling techniques with learning optimal threshold in BN to deal with imbalanced data. Concretely, we first rebalance the datasets by using resampling methods. We then learn the optimal threshold on the posterior probabilities produced by some Bayesian classifiers such as general BN, TAN, BAN, and Markov Blanket structure on those data. We optimize the threshold for each classifier on the holdout-set to maximize the F1-Measure, then using this threshold for final classification. We also make these classifiers become cost-sensitive by injecting the unequal misclassification costs to the threshold. Our experimental results show that the proposed methods significantly outperform the baseline Naive Bayes. These methods also perform as good as the state-of-the-arts and significantly better in certain cases.**

I. Introduction

In binary classification problems, class imbalance is a problem which the class distribution is far from the uniform distribution. This phenomenon appears in many machine learning applications, such as credit card fraud detection, intrusion detection, oil-spill detection, disease diagnosis, and many other areas ([1], [2], [3]). Most classifiers in supervised machine learning 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 problem that degrades the performance of such classifiers ([1], [2]). It is also considered as one of ten challenging problems in machine learning research [4].

Researchers have introduced many techniques to deal with class imbalance, as summarized in [1], [2]. These techniques can be categorized into 3 main groups: Pre-processing, internal processing, and post-processing. In the pre-processing group, most of them focus on (re)sampling methods such as in [5], [6]. In the internal classifier processing group, the algorithms are specifically designed for specific classifiers such as SVM [7], or C4.5 [8], or more [1], [2].

This work focuses on the last group – post-processing method, which mainly relies on the posterior probabilities produced by the classifiers. Many papers have been published in this area such as meta cost-sensitive learning (CSL) ([9], [10], [11]). Most of them have applied CSL to C4.5 ([10], [9]), Naive Bayes (NB) ([12]), and support vector machines [11]. Since the methods in this group are based on posterior probabilities, it is important to know that which classifiers that one choose should produce good probabilities. Moreover, as discussed in ([13], [14]), averaging on probabilities can produce results better than on voting the majority. Bayesian Network (BN) is a good candidate for this choice, but as far as we know, almost the literatures focused on SVM, C4.5, and Naive Bayes, which has a strong assumption on the independence among attributes given the class attribute. As shown in [15], relaxing on this assumption can lead to the better results. In this study, we propose methods which utilize the Bayesian posterior probabilities to deal with imbalanced data. Concretely, the contributions of this work are described as the followings:

- In the first method, we first rebalance the dataset by using sampling techniques[1]. We then optimize the decision threshold of the posterior probabilities produced by Bayesian Networks (e.g. general BN, TAN, BAN, and Markov Blanket structure[2]) to maximize the F1-Measure. Once the optimal threshold is archived, we use it for final prediction.
- In the second method, instead of optimizing the decision threshold on the holdout-set, we introduce a threshold from a ratio of unequal misclassification costs.
- We compare these methods with not only the baseline, but also the state-of-the-arts such as SMOTE and TOMEK LINK.

Experimental results show that the proposed methods significantly outperform the baseline, perform as good as the state-of-the-arts and significantly better in certain cases.

[1] We can use any sampling method but in this study we experiment on random undersampling, random oversampling, SMOTE [6], and TOMEK LINK [5]

[2] We will review these methods in section IV

II. Related Work

Previous works have tuned the decision threshold by some different ways. [16] has experimented on the moving of decision threshold of the ROC curve and adjusted the cost matrix to deal with unbalanced and unknown cost data. They compared the results of C5.0, Nearest Neighbor, and Naive Bayes; In [17], the authors used a method which varies from [10] called Thresholding. This method uses C4.5 as a base classifier and selects a proper threshold from training instances according to the misclassification costs. The results are reported in term of total cost and took the misclassification costs into account; [18] proposed an ensemble of Naive Bayes classifiers with an adjusted decision threshold trained on random undersampling data to deal with class imbalance; while [11] proposed combining sampling techniques with CSL.

Difference from the literature, we focus on BNs. At first, we rebalance the datasets, then we locally optimize the decision threshold (as well as employ the misclassification costs to the threshold) on the posterior probabilities produced by several BNs to deal with class imbalance.

III. Dealing with Class Imbalance

A. Main Techniques

To deal with imbalanced datasets, many techniques have been introduced as summarized in ([1], [2]). We will briefly introduce some techniques which are used in this study.

Random oversampling (**ROS**) is a non-heuristic method used to balance class distribution by randomly duplicating the minority class examples, while random undersampling (**RUS**) randomly eliminates the majority class examples.

Tomek's Link (**TLINK**) [5] is a method for cleaning data. Given two examples e_i and e_j belonging to different classes, $d(e_i, e_j)$ be the distance between e_i and e_j. A pair (e_i, e_j) is called a TLINK if there is no example e_l such that $d(e_i, e_l) < d(e_i, e_j)$ or $d(e_j, e_l) < d(e_i, e_j)$. If there is a TLINK between 2 examples, then either one of these is noise or both of them are borderline examples. We want to use TLINK as undersampling method, so only majority examples are removed.

The Synthetic Minority Oversampling Technique (**SMOTE**) is an oversampling method introduced by [6] which generates new artificial minority examples by interpolating between the existing minority examples. This method first finds the k nearest neighbors of each minority example; next, it selects a random nearest neighbor. Then a new minority class sample is created along the line segment joining a minority class sample and its nearest neighbor.

Another approach is cost-sensitive learning **(CSL)**, which takes the misclassification costs into account. Most classifiers assume that the misclassification costs are the same. In most real-world applications, this assumption is not true. For example, the cost of misclassifying a non-terrorist as terrorist is much lower than the cost of misclassifying an actual terrorist who carries a bomb to a flight; or in cancer diagnosis, misclassifying a cancer is much more serious than the false alarm since the patients could lose their life because of a late diagnosis and treatment [17]. Cost is not necessarily monetary, for examples, it can be a waste of time or even the severity of an illness [10].

This study focuses on binary classification problems. We denote the positive ($+$ or $+1$) as the minority, and the negative ($-$ or -1) as the majority. Let $C(i, j)$ be the cost of predicting an example belonging to class i when in fact it belongs to class j, then the cost matrix is defined in Table I-B.

TABLE I
A. Confusion matrix (left) and B. Cost matrix (right)

		Predict	
		$+$	$-$
Actual	$+$	TP	FN
	$-$	FP	TN

		Predict	
		$+$	$-$
Actual	$+$	$C(+,+)$	$C(-,+)$
	$-$	$C(+,-)$	$C(-,-)$

Given the cost matrix, the minimum expected loss can be determined by Bayes risk as in Eq. (1), where $P(j|x)$ is the posterior probability of classifying an example x as class j.

$$\mathcal{R}(i|x) = \sum_{j \in \{-,+\}} P(j|x)C(i,j) \tag{1}$$

Difference from [10], we do not take "profit" into account, so we assume that there are no costs (or profits) for correct classifications. The cost threshold is introduced as in the following:

Lemma 1: *Given the cost matrix (or cost ratio), a threshold for the classifier can be determined by* $\theta = \frac{C(+,-)}{C(+,-)+C(-,+)}$ *and a new example can be classified as positive if* $P(+|x) > \theta$.

Proof: To classify as positive, an example x must be satisfied $\mathcal{R}(+|x) < \mathcal{R}(-|x)$, substitute this expression by Eq. 1, we will have $P(-|x)C(+,-) + P(+|x)C(+,+) < P(-|x)C(-,-) + P(+|x)C(-,+)$. Since $C(+,+) = 0$ and $C(-,-) = 0$, this reduced to $P(-|x)C(+,-) < P(+|x)C(-,+)$, equivalent to $(1 - P(+|x))C(+,-) < P(+|x)C(-,+)$ lead to $P(+|x) > \frac{C(+,-)}{C(+,-)+C(-,+)}$, or $P(+|x) > \theta$ ■

B. Evaluation Metrics

When evaluating on imbalanced data, the accuracy metric becomes useless. For example, suppose a dataset has 990 negative and only 10 positive examples (these minorities are usually the interest one). Since most classifiers are designed to maximize their accuracy, in this case, they will classify all examples belong to the majority class to get the maximum of 99% accuracy. However, this result has no meaning because all the positive examples are misclassified. To evaluate the model in such case, researchers usually use F-Measure and the area under the ROC curve (AUC), which are related to some other metrics described in the following.

The Recall (R), also called true positive rate (TPR), is a proportion of positive examples correctly classified as belonging to the positive class, determined by $R = TP/(TP+FN)$. The Precision (P) is the positive predictive value determined by $P = TP/(TP + FP)$. F-Measure is an evaluation metric which considers both the Recall and the Precision (β is usually set equal to 1, called F1-Measure). F-Measure = $\frac{(1+\beta^2)\times P\times R}{(\beta^2\times P+R)}$

Another metric is GMean ($GMean = \sqrt{TNR \times TPR}$), which balances both true positive rate (TPR) and true negative rate (TNR). We will use these metrics to evaluate our models in section VI-B.

IV. Bayesian Network Classifiers

A. Bayesian Networks

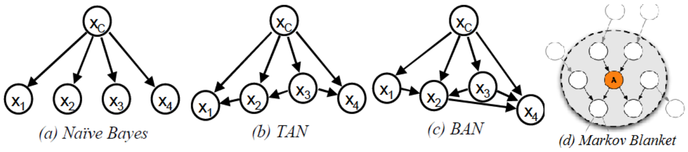

Fig. 1. Bayesian Network types

Bayesian Network (BN) is defined by a pair $B = \langle G, \Theta \rangle$, where G is the directed acyclic graph with a set of nodes $X = (x_1, x_2, \dots, x_n)$ represent random variables, and edges represent the direct dependencies between these variables, and Θ is a set of parameters of the network [15].

Naive Bayes **(NB)** is a type of BN which has assumptions that all the variables are conditionally independent given the class variable and are directly dependent on the class variable, as in Fig. 1a[3].

Tree Augmented Naive Bayes **(TAN)** [15] relaxes the assumption in NB by allowing arcs between the children of the target node, as in Fig. 1b.

Bayesian Network Augmented Naive Bayes **(BAN)** [20] is a BN which all other nodes are children of the target node, but a complete BN is constructed between the child nodes rather than just a tree as in TAN [19].

The Markov Blanket Bayesian Classifier **(MB)** [19] is a BN which has Markov Blanket property at a target node. The Markov Blanket for a node in BN consists of its parents, its children, and the parents of its children, as in Fig. 1d.

B. Learning in Bayesian Networks

Like other literatures [15], [20], [19], [21], this study focuses on the discrete and non-missing value variables[4]. The learning tasks in BN consist of two steps. The first step is to learn the network structure and the second step is to compute the conditional probability tables (CPTs).

To learn the structure B_S of the BN, we can consider it as an optimization problem [21] and need to maximize the quality measure Q of B_S given dataset D. In this study, we use the Bayesian metric as a quality measure, determined by the following equation:

$$Q(B_S|D) = P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{2}$$

where $P(B_S)$ is prior probability of B_S; n is the number of variables; Γ is a Gamma function; r_i and q_i are the cardinality of node x_i and a set of its parents Π_i, respectively; N_{ij} is $|D|$ for which Π_i takes its jth value; N_{ijk} is $|D|$ for which Π_i takes its jth value and for which x_i takes its kth value; $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$; N'_{ij} and N'_{ijk} represent choices of priors on counts restricted by $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$ [21]. Since $P(B_S)$ is constant, to maximize Q, we just need to maximize the second inner product in Eq. (2) as the following

$$\Psi = \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{3}$$

To do this, we use K2 algorithm [22] which initially assumes that a node has no parents, and then adding incrementally its parent that can increase the probability of the resulting network. This process repeats greedily until the addition of the parent does not increase the network structure probability. Concretely, each iteration of K2, an arc is added to node u from the node v that maximizes $\Psi(u, \Pi_u \cup v)$, where Π_u is the set of parents of node u. If $\Psi(u, \Pi_u) > \Psi(u, \Pi_u \cup v)$ then no arc is added [19].

After B_S is learned, the CPTs can be estimated by:

$$P(x_i = k | \Pi_{x_i} = j) = \frac{N_{ijk} + N'_{ijk}}{N_{ij} + N'_{ij}} \tag{4}$$

Once having the CPTs, one can infer for any new event. The probability of an arbitrary event X = $(x_1, x_2, \dots, x_n)$ is determined by

$$\mathcal{P}(X) = \mathcal{P}(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} \mathcal{P}\big(x_i \,\big|\, \Pi_{x_i}\big) \tag{5}$$

Given a dataset D consists of a target y and a set of attributes $X = (x_1, x_2, \dots, x_n)$, we can infer the class for y by calculating the $\arg\max_y(\mathcal{P}(y|X))$ from the probability distribution of Eq. (5)[5].

V. Proposed Methods

We propose 2 methods for dealing with imbalanced data. We implement these methods using WEKA[6].

- **Method 1**: We first rebalance the datasets by using sampling techniques (e.g. ROS, RUS, SMOTE, or TLINK). We then locally optimize the threshold θ to maximize the F1-Measure (maximize the *Recall* for the minority class but also taken into account the *Precision* to prevent the degradation of the overall model). Once having an optimal θ, we use it for the final classification. We experiment this method on several BNs such as general BN, TAN, BAN, and Markov Blanket BN classifier, which named as BN-Opt, TAN-Opt, BAN-Opt, and MB-Opt respectively.
- **Method 2**: Instead of optimizing the threshold, we use the cost threshold in Lemma 1 and learn on original data. This method is called BNCost.

[3] Picture sources: [19] and Wikipedia (en.wikipedia.org)

[4] We can discretize the numeric attributes, and replace all missing values for nominal and numeric attributes with the modes and means, respectively.

[5] The interesting readers can read the article [15] for details

[6] www.cs.waikato.ac.nz/ml/weka

- We compare these methods with not only the baseline Naive Bayes but also the state-of-the-art SMOTE and TLINK.

```
1: procedure LEARN-R-OPT-BNS($\mathcal{D}_{Train}, \mathcal{D}_{Test}, \Phi, \mu, \Theta$)
     Input: $\mathcal{D}_{Train}$ with $x_i \in \mathcal{X}$, target $y_i \in \{-1, +1\}$
       $\Phi \in$ {BN, TAN, BAN, BN Markov Blanket}
       $\mu \in$ {SMOTE, TLINK, ROS, RUS}
       $\Theta \in$ {Sampling percentages}
     Outputs: Label for new example $x^*$ in $\mathcal{D}_{Test}$
2:     for each $\delta \in \Theta$ do
3:         $\mathcal{D}_{Train\mu\delta} \leftarrow GenerateDistribution(\mathcal{D}_{Train}, \mu, \delta)$
4:         $\theta \leftarrow OptimizeThreshold(\mathcal{D}_{Train\mu\delta}, \Phi)$
5:         $\theta^* \leftarrow$ Update the best value of $\mathcal{D}^*_{Train\mu\delta}$
6:     end for
7:     Learn BN structures and CPTs as in Eq. (2, 3, 4)
         $\mathcal{P}(x_1, x_2, \ldots, x_n) \leftarrow \left( \prod_{i=1}^{n} \mathcal{P}\big(x_i \,\big|\, \Pi_i\big); x_i \in \mathcal{D}^*_{Train\mu\delta}; \Phi \right)$
8:     Test for new example $x^*$ from $\mathcal{D}_{Test}$
         $\mathcal{H}(x^*) \leftarrow \mathcal{I}\big(\mathcal{P}(j = +1 | x^*) > \theta^*\big)$
9: end procedure
```

Fig. 2. Learning BNs with optimal threshold on resampling data

The proposed methods are formulated in Fig. 2, called *Learn-R-Opt-BNs*. At first, we apply sampling technique on the train set to generate new datasets with more balanced distributions; then, we optimize the threshold of BN Φ to maximize the F1-measure as in lines 2-6. The optimal threshold θ^* and $\mathcal{D}^*_{Train\mu\delta}$ are recorded. The next steps are to learn the structure of that BN and compute the CPTs as in line 7. Once the CPTs are constructed, we can use them for inferring new examples in the test set as in line 8. The indicator function $\mathcal{I}(.)$ gives the positive class if the expression is true and negative class for the inverse. The θ^* value can be the optimal threshold for the first method or the cost threshold (Lemma 1) for the second method. Please note that in the second method, most datasets do not have the cost matrix (or cost ratio), so we assumed the cost ratio from the set $\{2^1, 2^2, 2^4, 2^6, 2^8\}$. (This assumption is also done in many other studies [9], [17], [11]). We locally search for the best ratio in this set and apply it to the final prediction as in [11].

In Fig. 3, we do 5-folds cross-validation on the holdout set to get the average prediction scores (line 4) and get the unique values from those scores (line 5) for the minority class. We then consider each score value as a threshold and recalcualte the F1-Measure. We update the threshold which has the maximal F1 value (line 7-13). We can treat this threshold as a hyperparameter and do the hyperparameter search as in ([3], [11]), but this method need more times than the current one.

```
1: procedure OPTIMIZETHRESHOLD($\mathcal{D}_{Tr}, \Phi$)
     Input: dataset $\mathcal{D}_{Tr}$, Bayesian Network $\Phi$
     Outputs: the best threshold for F1Measure (F1)
2:     $(\mathcal{D}_{LocalTrain}, \mathcal{D}_{Val}) \leftarrow \mathcal{D}_{Tr}$        ▷ split for 5-folds CV
3:     $\mathcal{M} \leftarrow buildLocalModel(\mathcal{D}_{LocalTrain}, \Phi)$
4:     $predictionScores \leftarrow testLocalModel(\mathcal{M}, \mathcal{D}_{Val}, \Phi)$
5:     $UniqueScores \leftarrow$ Unique-Values($predictionScores$)
6:     $bestF1 \leftarrow 0$; $bestThreshold \leftarrow 0$
7:     for each $curThreshold \in UniqueScores$ do
8:         $currentF1 \leftarrow$ Calculate $F1$ using $curThreshold$
9:         if ($currentF1 > bestF1$) then
10:            $bestF1 \leftarrow currentF1$
11:            $bestThreshold \leftarrow curThreshold$
12:        end if
13:    end for
14:    return $bestThreshold$
15: end procedure
```

Fig. 3. Optimize the threshold on the holdout-set

VI. Experimental Results

A. Datasets

We have experimented on 16 imbalanced datasets collected from the UCI repository[7], as described in Table II. Some multi-class datasets are converted to binary-class using one-versus-the-rest. We encode the class which has the smallest number of examples as the minority (positive) class, and the rest as the majority (negative) one. The imbalance ratio ranges from 1.77 to 64.03.

TABLE II
Datasets

Dataset	#Examples	#Attributes	#Minority	Imba. Ratio
Abalone	4,177	9	391	9.68
Allbp	2,800	30	133	20.05
Allhyper	3,772	30	102	35.98
Allrep	3,772	30	124	29.45
Ann	7,200	22	166	42.37
Anneal	898	39	40	21.45
Breastcancer	699	11	241	1.90
Diabetes	768	9	268	1.86
Dis	3,772	30	58	64.03
Heartdisease	294	14	106	1.77
Hypothyroid	3,163	26	151	19.95
IJCNN	49,990	22	4,853	9.70
Nursery	12,960	9	328	38.51
Pima-Indian	768	9	268	1.87
Sick	2,800	30	171	15.37
Transfusion	748	5	178	3.20

B. Results

We use the paired t-tests (2-tails) with significance level 0.05 for all the experiments. The results are averaged from 5-folds cross-validation. We do the hyperparameter search to determine the best cost ratio from the set $\{2^1, 2^2, 2^4, 2^6, 2^8\}$[8].

[7] http://archive.ics.uci.edu/ml/

[8] We point the interesting readers to the article [11] for more details

Tables III presents the detailed results of F1-Measure and averaged results of other metrics when using SMOTE as a sampling method in Learn-R-Opt-BNs. We report the results of the proposed methods together with 4 other classifiers: Naive Bayes, general BN, SMOTE, and TLINK without optimizing the threshold.

In the first experiment, we use Naive Bayes as a baseline. We can observe that all the remaining methods can win this baseline easily. From this experiment, we recognize that Naive Bayes does not work well. The reason could be because of its independent assumption, as discussed in the literature ([15], [19]). Because of this reason, in the remaining experiments, we use general BN as a base classifier[9] for SMOTE and TLINK. Three "w/t/l" rows mean that we check the significantly different results (win/tie/lose) of the remaining methods with the "base". Clearly, the proposed method MB-Opt, TAN-Opt, and BAN-Opt outperform the SMOTE 7 significant results out of 16 datasets (7/9/0). The MB-Opt gives the best average result of F1-Measure among the others.

Please note that in this work, we just optimized the results for F1-Measure, but for referencing, we also report the AUC, GMean, and Recall on average. The average results of AUC and GMean show that MB-Opt, again, is the best classifier among the others.

Table IV-A, IV-B, and IV-C show the results of using TLINK, ROS, and RUS as a sampling method in Learn-R-Opt-BNs, respectively. Because of limitation in the space, we just show the averaged results. The MB-Opt and BAN-Opt also outperform the other methods.

From these results, we recognize that the BAN-Opt can perform better if we remove "noise" and "borderline" examples by using TLINK or remove randomly by RUS, while the MB-Opt requires more artificial data generated by SMOTE or ROS to get better performance.

VII. Conclusion

This study introduces two methods which utilize the Bayesian posterior probabilities to deal with imbalanced data. We experiment these methods on several Bayesian classifiers such as general BN, TAN, BAN, and Markov Blanket BN classifier. In the first method, we rebalance the datasets by using sampling techniques. We then locally optimize the threshold to maximize the F1-Measure (maximize the *Recall* but also take into account the *Precision* to avoid the degradation of overall model). Once the optimal threshold is found, we use it for final classification. The second method injects the unequal misclassification costs to the threshold. Experimental results show that the proposed methods significantly outperform the baseline Naive Bayes. They also perform as good as the state-of-the-art methods and significantly better in certain cases. Thus, these methods can be good candidates for learning from imbalanced data. In future work, we will analyze more details on the results and compare the proposed methods when learning on original datasets.

[9] SMOTE and TLINK are sampling methods, so they need a base classifier

Acknowledgments

The first author was funded by the "TRIG - Teaching and Research Innovation Grant" Project of Can Tho University, Vietnam.

References

[1] 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.

[2] 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.

[3] N. Thai-Nghe, A. Busche, and L. Schmidt-Thieme, "Improving academic performance prediction by dealing with class imbalance," *in Proceeding of 9th IEEE International Conference on Intelligent Systems Design and Applications (ISDA09)*, 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] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man, and Communications SMC-6*, pp. 769–772, 1976.

[6] 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.

[7] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of International Joint Conference on Artificial Intelligent (IJCAI'99)*, 1999, pp. 55–60.

[8] 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*, 2010, pp. 766–777.

[9] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," *5th ACM SIG-KDD International conference on Knowledge Discovery and Data mining*, pp. 155–164, 1999.

[10] C. Elkan, "The foundations of cost-senstive learning," *17th International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

[11] 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 (IJCNN10)*, July 2010.

[12] V. S. Sheng, C. X. Ling, A. Ni, and S. Zhang, "Cost-sensitive test strategies," in *American Annual Conference on Artificial Intelligence (AAAI)*, 2006.

[13] W. Fan, E. Greengrass, J. McCloskey, P. S. Yu, and K. Drummey, "Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches," *IEEE International Conference on Data Mining*, pp. 154–161, 2005.

[14] S. Hido and H. Kashima, "Roughly balanced bagging for imbalanced data." in *Proceedings of the SIAM International Conference on Data Mining*, 2008, pp. 143–152.

[15] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[16] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequeal and unknown," *Workshop on Learning from Imbalanced Data Sets II, ICML*, 2003.

[17] V. S. Sheng and C. X. Ling, "Thresholding for making classifiers cost-sensitive," in *American Annual Conference on Artificial Intelligence (AAAI)*, 2006.

[18] W. Klement, S. Wilk, W. Michaowski, and S. Matwin, "Dealing with severely imbalanced data," *Workshop on Data Mining When Classes are Imbalanced and Errors Have Costs, PAKDD*, 2009.

[19] M. G. Madden, "A new bayesian network structure for classification tasks," in *Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*. London-UK: Springer-Verlag, 2002, pp. 203–208.

[20] J. Cheng and R. Greiner, "Comparing bayesian network classifiers," *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI-99)*, 1999.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Exploration*, vol. 11, no. 1, 2009.

[22] G. F. Cooper and T. Dietterich, "A bayesian method for the induction of probabilistic networks from data," in *Machine Learning*, 1992, pp. 309–347.

TABLE III

DETAILED RESULTS OF F1-MEASURE AND AVERAGE OF OTHER METRICS. THE BOLD NUMBER IS THE BEST VALUE IN EACH ROW

Dataset	NaiveBayes	BayesNet	SMOTE	TLINK	Learn-R-Opt-BNs (R is SMOTE)				
					BN-Opt	MB-Opt	TAN-Opt	BAN-Opt	BNCost
abalone	0.361±0.014	0.370±0.013 ○	0.379±0.018 ○	0.380±0.014 ○	0.410±0.020 ○	**0.416**±0.017 ○	0.410±0.018 ○	0.407±0.021 ○	0.380±0.013 ○
allbp	0.522±0.077	0.559±0.069 ○	0.598±0.070 ○	0.589±0.068 ○	0.563±0.080	0.579±0.049	0.586±0.075	0.559±0.055	**0.603**±0.065 ○
allhyper	0.490±0.071	0.519±0.086 ○	0.495±0.098	0.554±0.091 ○	0.548±0.108	**0.691**±0.066 ○	0.632±0.112 ○	0.663±0.058 ○	0.553±0.105 ○
allrep	0.407±0.066	0.519±0.067 ○	0.632±0.076 ○	0.652±0.049 ○	0.665±0.060 ○	**0.829**±0.056 ○	0.753±0.030 ○	0.757±0.049 ○	0.664±0.075 ○
ann	0.801±0.053	0.852±0.043 ○	0.901±0.035 ○	0.887±0.056 ○	0.921±0.023 ○	0.931±0.025 ○	**0.933**±0.029 ○	0.928±0.026 ○	0.907±0.037 ○
anneal	0.583±0.135	0.699±0.115 ○	0.874±0.078 ○	**0.920**±0.089 ○	0.833±0.108 ○	0.908±0.079 ○	0.881±0.074 ○	0.889±0.109 ○	**0.920**±0.089 ○
breastcancer	0.944±0.011	0.952±0.010	0.959±0.014	0.962±0.011 ○	**0.964**±0.011 ○	0.956±0.018	0.948±0.018	0.949±0.012	0.960±0.014
diabetes	0.645±0.089	0.646±0.082	0.651±0.054	0.662±0.071	**0.668**±0.025	0.667±0.046	0.667±0.054	0.670±0.044	0.646±0.077
dis	0.276±0.031	0.362±0.018 ○	0.397±0.075	0.477±0.075 ○	0.398±0.057 ○	0.300±0.120	0.418±0.095 ○	0.325±0.107	**0.521**±0.085 ○
heartdisease	**0.795**±0.062	0.770±0.062	0.753±0.096	0.750±0.075	0.742±0.096	0.763±0.078	0.746±0.090	0.749±0.089	0.743±0.083
hypothyroid	0.778±0.035	0.826±0.027 ○	0.841±0.034 ○	0.867±0.042 ○	0.832±0.030 ○	0.837±0.062	0.856±0.046 ○	0.850±0.030 ○	**0.871**±0.032 ○
ijcnn	0.304±0.019	0.359±0.020 ○	0.515±0.015 ○	0.415±0.023 ○	0.538±0.005 ○	**0.653**±0.006 ○	0.579±0.015 ○	0.635±0.015 ○	0.410±0.025 ○
nursery	0.380±0.035	0.384±0.034	0.680±0.039 ○	0.548±0.021 ○	0.738±0.022 ○	**0.897**±0.026 ○	**0.897**±0.026 ○	**0.897**±0.026 ○	0.387±0.035
pima	0.634±0.089	0.635±0.078	0.635±0.066	0.647±0.081	**0.653**±0.038	0.645±0.055	0.639±0.054	0.642±0.047	0.636±0.073
sick	0.554±0.071	0.642±0.063 ○	0.703±0.093 ○	0.746±0.047 ○	0.762±0.085 ○	**0.798**±0.064 ○	0.776±0.066 ○	0.779±0.062 ○	0.757±0.052 ○
transfusion	0.282±0.033	0.404±0.014 ○	0.491±0.026 ○	0.486±0.022 ○	0.450±0.026 ○	**0.492**±0.014 ○	**0.492**±0.014 ○	**0.492**±0.014 ○	0.488±0.027 ○
F1 Average	0.547	0.594	0.657	0.659	0.668	**0.710**	0.701	0.700	0.653
w/t/l	base	11/5/0	10/6/0	13/3/0	11/6/0	9/7/0	11/5/0	10/6/0	11/5/0
w/t/l	0/5/11	base	7/8/1	11/5/0	8/8/0	9/7/0	9/7/0	10/6/0	11/5/0
w/t/l	0/6/10	1/8/7	base	3/11/2	3/13/0	7/9/0	7/9/0	7/9/0	3/11/2
AUC average	0.905	0.909	0.915	0.917	0.915	**0.923**	0.920	0.921	0.917
GMean Average	0.754	0.773	0.813	0.802	0.816	**0.831**	**0.831**	0.829	0.789
Recall average	0.649	0.668	0.728	0.717	0.778	0.784	**0.787**	0.786	0.688

Paired t-tests with significance level 0.05; ○, ● statistically significant improvement or degradation; w/t/l compares the number of significant win/tie/lose times when compared to the "base"

TABLE IV

AVERAGED RESULTS OF F1-MEASURE AND OTHER METRICS. A. TLINK AS SAMPLING METHODS IN LEARN-R-OPT-BNS

Dataset	NaiveBayes	BayesNet	SMOTE	TLINK	Learn-R-Opt-BNs (R is TLINK)				
					BN-Opt	MB-Opt	TAN-Opt	BAN-Opt	BNCost
F1 Average	0.547	0.594	0.657	0.659	0.683	**0.728**	0.711	**0.728**	0.653
AUC Average	0.905	0.909	0.915	0.917	0.917	**0.922**	0.921	0.921	0.917
GMean Average	0.754	0.773	0.813	0.802	0.820	0.840	0.825	**0.844**	0.789
Recall Average	0.649	0.668	0.728	0.717	0.779	0.792	0.775	**0.798**	0.688

B. ROS as sampling methods in Learn-R-Opt-BNs

Dataset	NaiveBayes	BayesNet	SMOTE	TLINK	Learn-R-Opt-BNs (R is ROS)				
					BN-Opt	MB-Opt	TAN-Opt	BAN-Opt	BNCost
F1 Average	0.547	0.594	0.657	0.659	0.679	**0.719**	0.705	0.715	0.653
AUC Average	0.905	0.909	0.915	0.917	0.918	**0.925**	0.923	0.923	0.917
GMean Average	0.754	0.773	0.813	0.802	0.832	0.839	0.841	**0.846**	0.789

C. RUS as sampling methods in Learn-R-Opt-BNs

Dataset	NaiveBayes	BayesNet	SMOTE	TLINK	Learn-R-Opt-BNs (R is RUS)				
					BN-Opt	MB-Opt	TAN-Opt	BAN-Opt	BNCost
F1 average	0.547	0.594	0.657	0.659	0.675	**0.720**	0.708	0.716	0.653
AUC Average	0.905	0.909	0.915	0.917	0.917	**0.922**	**0.922**	**0.922**	0.917
GMean Average	0.754	0.773	0.813	0.802	0.825	0.847	0.844	**0.852**	0.789