哈尔滨工业大学（深圳）

Harbin Institute of Technology, Shenzhen

Interim Assessment of the Thesis for the Master's Degree

Name	Ying Zhou
Entrance Date	2016.9.1
Thesis Title	Improvement of generative model
and its application in imbalanced classification	
Discipline	Computer Science
Supervisor	Chunkai Zhang
Report Date	2018.4.28

1. Does the thesis progress according to the research objectives and schedule as stated in the primary report? (at least 100 words)

In this study, I proposed a new distribution-based oversampling method, using a variational auto-encoder to model the distribution of the minority class samples, and randomly sample the model to generate more reasonable synthetic samples. The experimental results prove the effectiveness of the algorithm. At the same time, a new evaluation standard of the data set imbalance degree is proposed. The new evaluation of the data set is considered from the inconsistency of classification difficulty for the different class samples, and it’s an improvement of the traditional sample set size imbalance ratio. The new evaluation has a highly linear correlation with the final classification result. The evaluation is used to filter the randomly sampled samples in the previous step to improve the final recognition performance for the minority class.

2. The completed work and its related outcomes (at least 1500 words).

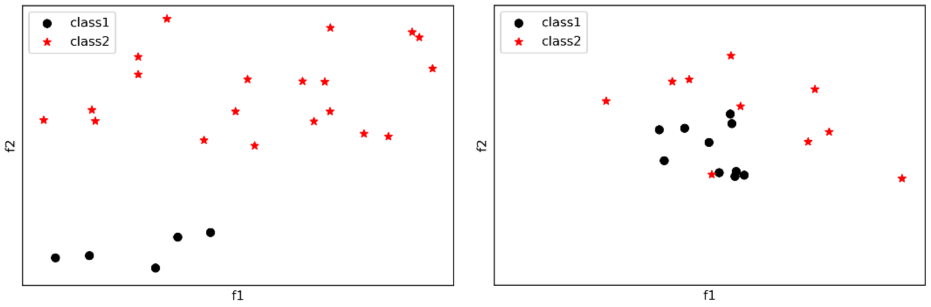
In this work, given a dataset X which has two classes: first denote the positive data by P:$\{x_1,x_2,\cdots x_{N_+}\}$, which are samples from an unknown distribution F, and the negative dataset by N:$\{x_{N_+ +1}, x_{N_+ +2},\cdots, x_N\}$, which are sampled from an unknown distribution G, then we have X= PUN, suppose there are N- samples in N, and $N_+ + N_- =N$

The traditional imbalance ratio is defined as:

$$ir=\frac{N_{=}}{N_{+}} \tag{2-1}$$

When the positive and negative samples are distributed evenly, the imbalance ratio is useful in evaluating the classification difficulty of a dataset. However when it comes to the case in figure 2-1, when F is different from G, the imbalance ratio in figure 2-1(a) is 4.4 while it is 1.1 in figure 2-1(b), from the perspective of imbalance ratio, it should be more difficult to classify the data set in (a) than that in (b), but the data in (a) has clear linear boundaries, so in the same linear model, the sample in (b) can't get a 100% classification result. Therefore, in fact, the data set in (b) is more difficult to classify than in (a), which is contrary to the comparison result of imbalance ratio, the complexity of the distribution of the dataset cannot be reflected in imbalance ratio.

In this work, I proposed a generalized imbalance ratio. The classification difficulty of each sample is determined by the label of its neighbors. If the neighbors of a sample contain more samples in the different class, the sample becomes more difficult to determine its label. On the contrary, if a sample is surrounded by samples with the same label, it’s easier to determine its own label; the IGIR definition formula is as follows:



(a) (b)

Fig 2-1 the dilemma of IR

$$T_{+}=\frac{1}{N_{+}}\sum_{x\in P}\frac{1}{k\sum_{r=1}^{k} Ir(x,X)}=\frac{1}{N_{+}}\sum_{x\in P} t_k(x) \tag{2-2}$$

$$wei\text{-}T_{+}=\frac{1}{N_{+}}\sum_{x\in P} 1/k\sum_{r=1}^{k} weight*Ir(x,X)=\frac{1}{N_{+}}\sum_{x\in P} weight*t_k(x) \tag{2-3}$$

$$\mathrm{T}_{-}=\frac{1}{\mathrm{N}_{-}}\sum_{\mathrm{x}\in\mathrm{N}} \mathrm{t}_{\mathrm{k}}(\mathrm{x}) \tag{2-4}$$

$$wei\text{-}T_{-}=\frac{1}{N_{-}}\sum_{x\in N} weight*t_k(x) \tag{2-5}$$

$$GIR=T_{-}\text{-}T_{+} \tag{2-6}$$

$$IGIR=\sqrt{wei\text{-}T_{-}*wei\text{-}T_{+}} \tag{2-7}$$

The Ir(x,X) is a signal function, it gets 1 when x has the same label with its neighbor, and it gets 0 in contrast.

```
Algorithm 1: Computing the IGIR
  Input: A dataSet $X$,label $Y$,number of nearest neighbors $k$ in $k$-NN
  Output: IGIR
for $x$ in $X$ with label $y_x$ do
    $M \leftarrow$ the $k$ nearest neighbors of $x$
    $t_k(x) \leftarrow \frac{1}{k}\sum weight * IR(x, M)$
end
$wei - T_- \leftarrow \frac{1}{N_-}\sum t_k(x) * sgn(y_x == 0)$
$wei - T_+ \leftarrow \frac{1}{N_+}\sum t_k(x) * sgn(y_x == 1)$
IGIR $\leftarrow \sqrt{wei - T_- * wei - T_+}$
return IGIR
```

In the experimental results, I compare different classification results for several datasets with their corresponding evaluations. The measurements and classification results are shown in table 2-1. From the results, it can be seen that igir has greater improvement than previous ir and gir evaluation, and has a very high correlation with the final classification results. In addition, the classification results from different classifiers also have different correlations with the same indicator, the reason is that the different characteristics of these datasets can satisfy the different classification assumptions of these classifiers. In the SGD classifier (the linear regression classifier), the igir reflects the clarity of classification boundary between different class samples. The higher the IGIR is, the clearer the classification boundary is, and the better the classification result of the classifier is.

Table 2-1 Measurements and classification results

	IR	GIR	IGIR	F1_min	Gmean	Sensitivity
breasttissue	4.05	0.30	0.46	0.78	0.83	0.83
breastw	1.90	0.05	0.57	0.91	0.92	0.88
diabetes	1.87	0.22	0.37	0.57	0.66	0.57
german	2.33	0.37	0.33	0.47	0.60	0.48
glass	3.20	0.19	0.53	0.75	0.80	0.75
haberman	2.78	0.43	0.32	0.24	0.35	0.30
ionosphere	1.79	0.39	0.46	0.84	0.87	0.82
movement	14.00	0.44	0.47	0.59	0.77	0.69
satimage	8.15	0.04	0.59	0.95	0.97	0.94
Segment*	6.32	0.04	0.58	0.97	0.98	0.97
sonar	1.14	0.16	0.47	0.59	0.58	0.62
spect	3.85	0.61	0.32	0.50	0.65	0.51
vehicle	3.25	0.10	0.54	0.88	0.92	0.89
vertebral	2.10	0.14	0.47	0.67	0.71	0.66
wpbc	3.21	0.39	0.32	0.42	0.56	0.46
yeast0	5.08	0.38	0.41	0.49	0.68	0.53
yeast1	2.46	0.34	0.37	0.48	0.62	0.50
yeast2	2.21	0.26	0.37	0.49	0.61	0.49
yeast6	8.10	0.33	0.47	0.69	0.82	0.71

Table 2-2 R^2 of measurements and classification results

	F1_min	Gmean	sensitivity
IGIR	**0.92**	**0.88**	**0.93**
IR	0.18	0.34	0.28
GIR	-0.70	-0.58	-0.67

Where the $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2}$ and can be described as the relationship between X and Y, R^2 reflects how many percentages of the fluctuation of Y can be described by the fluctuation of X. That is to say, what percentage of the variance of the representation variable Y can be explained by the controlled variable X.

In this paper, aiming at the problem of imbalanced classification, I proposed a novel oversampling based on the idea of distribution-based oversampling: a generation model using the variational auto-encoder (VAE): taking VAE to model the probability distribution function and sampling the hidden layer space z to generate the final minority class synthesize samples, and improve the classification performance of the minority class. The network structure in this article is shown in Figure 2-2. Since the original VAE is applied to image generation, the synthesis results can be naturally visualized so as to determine their quality; however, the sequence data cannot be used directly in this article, and therefore needs some improvements:

Because there may be discrete features in the sequence data, but the random gradient descent used in VAE makes the generated features to be continuously differentiable, so before we start training VAE, this part of features needs to be removed and generated by other way.

Due to the small number of samples, it is impossible to reliably determine whether a feature is discrete or not. Therefore, there is an assumption that if a feature has fewer than 2 values, it is discrete. In fact, if a certain feature has only one value, this feature has no effect on the classification.

Before the process of training the VAE model, the feature number $nelements_j$ that appears in the j-th feature of the training set is first counted, excluding the discrete features, the formula is as follows:

$$nelements_j=\sum_{i}^{N_+} distinct\left\{x_{ij}\right\},1\leq j\leq d \tag{2-8}$$

$$x_i=\left\{x_{i1},x_{i2},\cdots,x_{ik}\right\}\cup\{x_{i(k+1)},\cdots,x_{id}\}$$

$$\text{s.t.}\begin{cases} nelements_j>2,\ 1\leq j\leq k \\ nelements_j\leq 2,\ \ k+1\leq j\leq d \end{cases} \tag{2-9}$$

If $nelements_j \leq 2$, the j-th column feature is a discrete feature, and conversely, it is a continuous feature. The features in the dataset are divided into continuous features and discrete features in order, and the continuous features are extracted as the final training set.

$$\text{XtrainVAE}=\begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{N_+1} & \cdots & x_{N_+k} \end{bmatrix} \tag{2-10}$$

Training a VAE model with XtrainVAE and sampling it randomly. Let the synthetic sample be Xnew, and Xfinal is the final composite sample, and the final training set is X∪ Xfinal.

$$\begin{cases} Xfinal_{ij}=Xnew_{ij}\cup X_{lm},k+1\leq m\leq d \\ s.t.\ agrmin\sum\left\|Xnew_{ij}-X_{lj}\right\|^2,1\leq j\leq k \end{cases} \tag{2-11}$$

The whole algorithm is shown in Algorithm 2.

Algorithm 2：VAEOS：VAE-based oversampling approach for imbalanced learning.

```
Input: X: dataset $X = P \cup N$ with N samples, consisting of $N_+$positive samples in P and $N_-$ negative samples in N
Output: Classifier H trained with dataset after the oversampling algorithm
Procedure:
1. Divide X into training dataset Xtrain and testing dataset Xtest
2. Data preprocessing according to formula (2-12)
3. Compute $\mathrm{nelements_j}$ for each feature j in Xtrain, decide each discrete feature and continuous feature
4. Decide XtrainVAE according to formula (2-10)
5. Use XtrainVAE to train a VAE model and randomly sample with the corresponding model
6. Synthesize the Xfinal according to formula (2-11)
7. Train a classifier H with Xtrain ∪ Xfinal
```

In the data preprocessing, the experimental data in this work comes from the UCI machine learning database. Some of them are multi-class data sets. In order to pursue high imbalance rates, we select one of the class samples as the minority class, and the rest of the samples are regarded as the majority. And for the missing value in the dataset, to ensure the integrity of the dataset, we use the most frequent value as a supplement to the missing attribute. We have used normalization to scale them. The formula is as follows:

$$x_{inew}=\frac{x_i\text{-}\bar{x}}{s} \tag{2-12}$$

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n} x_i\ ,s=\sqrt{\left(\frac{1}{n\text{-}1}\sum_{i}^{n}(x_i\text{-}\bar{x}\right)^2}$$

In this work, I use F1-value and gmean to evaluate the classification performance, the definitions are as follows:

$$F\text{-}value=\frac{(1+\beta^2)\times recall\times precision}{\beta^2\times recall+precision} \tag{2-13}$$

β is set to be 1.

$$gmean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} \tag{2-14}$$

In this work, the distribution-based over-sampling algorithm NDO and the classical interpolation algorithm SMOTE are compared. The classifier adopts naive Bayes in order to reduce the influence of parameters on the classification performance. In cross-validation, the minority and majority classes are simultaneously segmented to ensure that the data distribution is consistent with the original distribution. In order to reduce the influence of randomness on the final results, each algorithm calculated the average classification result of 10 runs of 10-fold cross-validation.

Table 2-3 F1-min for each dataset

	100%			200%			300%		
	VAE	NDO	SMOTE	VAE	NDO	SMOTE	VAE	NDO	SMOTE
1	94.22	**94.38**	**94.38**	**94.99**	94.38	94.38	**94.99**	94.38	94.38
2	**58.07**	55.66	56.26	**58.75**	56.63	56.45	**59.36**	56.27	56.45
3	**66.49**	65.47	62.44	**69.56**	66.55	61.15	**70.86**	61.90	61.15
4	**66.61**	65.93	66.27	**67.99**	66.74	66.33	**66.58**	65.59	66.33
5	**87.02**	82.34	80.54	**87.62**	82.63	82.71	**86.20**	81.44	82.71

From the comparison results of different oversampling algorithms, the oversampling algorithm proposed in this work can generate more reasonable samples, and the traditional oversampling algorithm will sacrifice the recognition performance of major class, while the algorithm proposed in this work can help classify both classes and improves the overall recognition effect.

Table 2-4 F1-maj for each dataset

	100%			200%			300%		
	VAE	NDO	SMOTE	VAE	NDO	SMOTE	VAE	NDO	SMOTE
1	96.77	**96.89**	**96.89**	**97.22**	96.89	96.89	**97.21**	96.89	96.89
2	**74.40**	72.06	72.35	**76.43**	71.98	71.87	**78.22**	71.73	71.90
3	91.54	**91.79**	89.69	**92.81**	92.22	89.41	**93.30**	92.47	89.03
4	**80.30**	79.73	80.25	**78.12**	77.78	76.71	**76.60**	74.95	74.48
5	**93.53**	89.62	87.79	**93.98**	89.84	88.56	**93.49**	90.07	89.45

Table 2-5 gmean for each dataset

	100%			200%			300%		
	VAE	NDO	SMOTE	VAE	NDO	SMOTE	VAE	NDO	SMOTE
1	96.04	**96.35**	**96.35**	**96.47**	96.35	96.35	**96.45**	96.35	96.35
2	**75.00**	72.71	73.27	**75.79**	73.50	73.18	**76.31**	73.30	73.34
3	**90.60**	89.55	88.91	**91.47**	89.23	89.36	**91.78**	89.41	89.01
4	**74.04**	73.57	73.84	**74.89**	74.07	73.01	**73.66**	73.24	73.03
5	**88.83**	86.47	85.32	**89.08**	86.66	85.99	**87.81**	86.86	86.98

3. The work to be completed and its schedule.

Adding igir as an evaluation index to help select the synthetic samples rather than a random sampling.

Improvements to VAE, because training VAE with a small number of class samples alone is not sufficient to train a strongly generative model. The cVAE architecture uses the minority and majority class samples as a training set for the model to help model the distribution of training samples. At the same time, improve the cVAE-gan architecture and mitigate the impact of data imbalance on the model.

The following schedule is as follows:

May 2018 - July 2018: Coding, implementation and testing, analysis of experimental results, and draw conclusions.

July 2018 - September 2018: Conduct overall system testing and summarize.

September 2018 - December 2018: Summarize the conclusions of the study, write a dissertation, and prepare for a reply.

4. The existing or expected difficulties and problems.

Igir is defined as the average number of k-nearest neighbors with the same label in the entire sample set. Therefore, it needs to calculate the distance for the entire data set to obtain its k-nearest neighbors. It takes a long time to calculate the k-nearest neighbors, and in the evaluation of the subsequent synthetic samples, it also needs to traverse the entire data set, so this part needs to be optimized. It is better to delete the synthesized sample that exceeds the original distance and reduce the running time of the algorithm.

In the oversampling part, due to the limited number of the minority class samples, add majority class samples in the training set may make sense, add the confrontation idea of the GAN model and construct an improved cVAE model, improve the quality of the synthesized samples, and alleviate the imbalanced dataset on the generative model.

5. The considerations on the possibility of completing the thesis on-time (at least 100 words).

In this work, I have now completed the algorithm of oversampling using VAE, and conducted the experiment to prove its effectiveness. Next, in addition to getting more experimental results, it is also necessary to optimize the oversampling algorithm based on VAE, in order to improve the quality of the synthesized sample and the final classification effect.

In the selection of the sample, the sample is measured using igir and its definition has been completed. In the subsequent work, the two solutions can be integrated.

In summary, the paper can be completed on schedule and achieve certain research results.

Supervisor’s Signature: