Handling Imbalance Visualized Pattern Dataset for Yield Prediction

Megat Norulazmi Megat Mohamed Noor
Graduate Dept of Computer Science, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia.
s91447@ss.uum.edu.my

Shaidah Jusoh
Graduate Dept of Computer Science, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia.
shaidah@uum.edu.my

Abstract

The prediction of the yield outcome in a non close loop manufacturing process can be achieved by visualizing the historical data pattern generated from the inspection machine, transform the data pattern and map it into machine learning algorithm for training, in order to automatically generate a prediction model without the visual interpretation needs to be done by human. Anyhow, the nature of manufacturing process dataset for the bad yield outcome is highly skewed where the majority class of good yield extremely outnumbers the minority class of bad yield. Comparison between the undersampling, over- sampling and SMOTE + VDM sampling technique indicates that the combination of SMOTE + VDM and undersampled dataset produced a robust classifier performance capable of handling better with different batches of prediction test data sets. Furtherance, suitable distance function for SMOTE is needed to improve class recall and minimize overfitting whilst different approach on the majority class sampling is required to improve the class precision due to information loss by the undersampling.

1. Introduction

Typically, [1] manufacturing processes such as hard disk media industries implementing process control through sampling rather than close loop system in each of its process equipment. The yield outcome of the manufacturing process will be determined by the inspection machine at the end of the process before they are packed and shipped.

In our previous work, [1] we introduced the approach to transform inspection machine generated numerical data from the nature that the data only can be used to learn the inspection machine behavior into a binary polynominal discretized data that will be able to be trained to predict "one step ahead" of the manufacturing yield outcome, whether it will be good or bad yield. As suggested by [2] shows that the proactive type of predictive maintenance method improves the efficiency of the maintenance, optimizes the maintenance planning and reduces the usage of resources such as labor and materials.

The result from our previous work [1] indicates that the combination of KStar learning algorithm with 12 bit binary polynominal decretized datasets giving the best result of class precision and recall compare to LWL, IBk learning algorithm and other discretized bit value of 4,6,8,10,14 and 16. However, even though KStar produced the best prediction accuracy result compare to other algorithm, we saw that the class recall for the *BAD* yield still not achieving significant improvement, not able to surplus 30% accuracy even with higher bit value discretized datasets.

Thus, combination with [3] SMOTE (Synthetic Minority Over Sampling Technique), [4] [5] random over sampling minority class and random under sampling majority class technique will be applied in this paper to improve the Bad yield class recall drawback issue without significantly affecting the class precision and overfitting issue.

2. Related work

What actually occurred with our previous work had been explained by [6] where a classifier induced from an imbalanced data set has typically a low error rate for the majority class and an unacceptable error rate for the minority class. The problem occurs when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class.

Random over-sampling that randomly replicates the minority class and the random under-sampling that

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

removes majority class instances was applied by [4] [5] in order to obtain a balanced distribution. However, as mentioned by [4] [5] [10], random under-sampling did not provide significant improvement over the original data set whereby random over-sampling was able to reduce significantly the FN rate, but it also increased the FP rate. Both have known drawbacks because under-sampling will cause lost of potentially valuable information from the removed instances where else over-sampling will increase the likelihood of over fitting issue due to the methods of making exact copies of the minority class examples.

However, despite of its limitation, [4] emphasized that over-sampling technique have the advantage because there is no information loss incur as what will potentially occur from under-sampling technique but in contrast with higher computational cost. Furthermore, [4] also highlighted that the over-sampling and under-sampling combination did not provide significant improvement if compare to the over-sampling alone.

In order to apply the advantage and minimize the disadvantage of over-sampling technique in handling with imbalance dataset, [3] [6] [7] proposed the SMOTE method which is an over-sampling technique by synthetically creates the instances rather replicates the exact copies from the minority class examples. The SMOTE and combination of SMOTE and under-sampling as proposed by [3] which are performed using C4.5, Ripper and a Naive Bayes classifier, performs better over other previous re-sampling method. SMOTE forces a bias towards the minority class because the synthetically generated instances cause the classifier to create generalize and less specific decision regions as compare to the replication of minority instances which creates a very specific decision region and leading to overfitting issue.

SMOTE over-sampling [3] application claimed to yield results by obtained the lowest FN rate, 2.50%, but also the highest FP rate, 15.24%. Compare with random oversampling, which present a 200% improvement in FN rate, with an increase of the FP rate in approximately 21%.

3. Approach Technique

3.1 Random Under-sampling and Over-sampling

The implementation of these non-heuristic approaches is very simple. We generated new dataset for training from original dataset by randomly pickup instances for under-sampling the majority class instances, over-sampling the minority class instances and combination of both under-sampling and over-sampling with specified level of sampling percentage.

3.2 SMOTE with VDM technique

SMOTE technique [3] was proposed to over-sample the minority class by selecting *k* minority class nearest neighbor instances and producing synthetic instances. Depending on the percentage of over-sampling required, neighbors from the *k* nearest neighbors are randomly chosen and the synthetic instances were generated by calculating the nearest neighbor numerical dataset with Euclidean distance function.

Since in our study that we are dealing only with nominal
value dataset generated by our novel data transform technique [1], we applied SMOTE over-sampling technique with modified Value Distance Metric (VDM) distance function as suggested by [3] to measure and obtain the k nearest neighbor instances. In our case we are using k=5 and to k=1, meaning that we were selecting 5 and 1 nearest neighbor instances from minority class dataset to generate a synthetic instance. Total numbers of synthetic instances were generated according to the number of over-sampling percentage required in our experiment.

The Value Difference Metric (VDM) distance δ, [6] between two corresponding feature values is defined as follows.

$$\delta(V_1, V_2) = \sum_{i=1}^{n} \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right|^k \quad (1)$$

Above equation indicates that, $V1$ and $V2$ are the two feature values. $C1$ is the total number of occurrences of feature value $V1$, and $C1i$ is the number occurrence of feature value V1 for classes i. $C2$ is the total number of occurrences of feature value $V2$, and $C2i$ is the number occurrence of feature value V2 for classes i. k is a constant, normally set to 1. The equation is used to calculate the value differences for each nominal feature in the given set of feature vectors.

As in our study, SMOTE-VDM was not used for classification purposes, i is equal to 1 because we only focus on minority class instances to produce new synthetic instances. To generate new minority class instances, [3] proposed to create new set instances values by taking the majority vote of the feature vector in consideration from its k nearest neighbors. Below shows an example of creating a synthetic instance by majority vote proposed.

Let F1 = P234 P1112 P3345 P975 P335 be the instance under consideration and let its 5 nearest neighbors be:-

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

F2 = P675 P678 P2341 P1234 P2334
F3 = P234 P789 P2242 P3345 P2334
F4 = P776 P456 P3456 P987 P567
F5 = P1234 P3567 P1112 P3345 P453
F6 = P234 P1112 P3345 P765 P777

The application of SMOTE-Nominal would create the following instance:

FS = P234 P1112 P3345 P3345 P2334

However, since we are dealing with polynominal data value and not with normal nominal value, the polynominal value which was discretized by 12 bit number producing 4096 possibility of nominal value for each attribute. Hence, there is a possibility that the majority vote technique to generate the synthetic value may not be feasible. This potentially was due to the high possibilities that there were no redundant pattern available for voting from the selected k nearest neighbors. Thus, we included an option in the VDM distance function by calculating the average of those 5 nearest neighbor selected in the instances attributes by converting the polynominal pattern into integer and then transform the calculated average number back to polynominal value. Anyhow, in the case of k equal to 1, the synthetic instances were generated directly from the selected nearest neighbor instances.

The threshold for VDM distance value in this study is 0.1. The VDM distance algorithm generates k nearest instance if the distance between two feature vectors which was randomly selected is less than 0.1. Zero is the ideal distance for similarity feature vector value but it is computationally expensive.

3.3 Performance Measure

As proposed by [4] we used F-measure to measure the overall performance of the sampled datasets studied. F-measure is a harmonic mean between recall and precision defined as:-

$$F = 2 \times \left(\frac{R \times P}{R + P} \right) \qquad (2)$$

The F-measure becomes zero if either R or P is zero and it will become 1 when both R and P are 1. R is recall and P is precision. Recall and precision are define as:-

$$R = \frac{\mathrm{CP}}{\mathrm{TP}} \qquad (3)$$

$$P = \frac{\mathrm{CP}}{\mathrm{PP}} \qquad (4)$$

CP is the number of instances that are correctly predicted as positive and TP is the number of actual positive instances, where PP is total number instances predicted as positive.

4. Study results

4.1 Procedure

We were using the same data from our previous study [1] where data fields used for the study were *ID*, *Total Yield Percentage, RankA, G-NG, R-NG, Ring, Hit, MP1, MP2, MP3, Q3MP3* and *Yield* class instance. 12 bit polynominal discretized training data was used as the original dataset to generate the new training data with random under-sampling, oversampling, undersampling + oversampling, SMOTE oversampling and SMOTE oversampling + random undersampling.

For plain random undersampling training dataset, they were generated by 30%, 40%, 50%, 60%, 70% and 80% from the original majority class dataset. We created 50%, 100%, 150%, 200%, 250% and 300% training datasets for random oversampling as well as SMOTE oversampling from the original minority class dataset. As for the combination sampling of random oversampling + undersampling and SMOTE + random undersampling datasets, the datasets were created by oversampling the original dataset minority class by 50%, 100%, 150%, 200%, 250% and 300% and then randomly undersampling the original majority class instances until it is reached to the same number of oversampled instances, so that their distribution will be exactly balanced.

Training datasets been trained with KStar algorithm as recommended [1] for the learning process with confusion matrix and stratified 10-fold cross validation. Classifiers generated from the training data were then being used to be tested with test data from the same batch with training data for the prediction test. The classifiers once again been tested with another test data from different batches to test the robustness of the generated classifiers.

4.2 Result analysis

Training result in table 4.1 shows that random undersampling the majority class instance was not giving significant improvement to the class recall and precision compare with original data set. Oversampling

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

results indicates that by oversampling the minority class instances randomly, the class recall increases proportionally with sampling percentage without significantly affecting the class precision. Hence, the F value shows significant improvement proportionally with higher number of the minority class instance oversampling. The combination of balance random over and undersampling result shows that the class recall increases proportionally with the number of sample but inconsistently affecting the precision. Even though the F value shows significant improvement compare to the original data set, oversampling minority class instances were producing the best result between these 3 sampling method.

Table 4.2 shows the training result of proposed technique SMOTE+VDM with nearest neighbor k=5. The result indicates that the sampling method was not able to improve the class recall. The result was even worst than the original datasets training outcome. Combination SMOTE+VDM with k=5 and undersampling shows better performance. However, the result still not able to overwhelm the plain oversample and over + undersampling method outcome.

Anyhow, table 4.3 explained that the SMOTE+VDM with k=1 result indicates better than k=5 for both with or without undersampling the majority class instances. The SMOTE+VDM and k=1, with and without undersampling both shows better performance from each other at certain condition of sampling percentage. However, the result still
underperforms the simple plain random oversampling method.

Table 4.4, 4.5 and 4.6 shows the result of the prediction test with classifiers generated by the training data. The results were based from the best performer classifiers selected from each sampling method. Table 4.4 indicates the

Table 4.1. Training result for undersampling, oversampling and its combination

Data Sets	Performance		
	R	P	F
Original data set	0.273	0.976	0.427
Undersampling			
30%	0.273	0.990	0.428
40%	0.265	0.975	0.417
50%	0.274	0.922	0.422
60%	0.281	0.945	0.434
70%	0.257	0.887	0.399
80%	0.282	0.777	0.414
Oversampling			
50%	0.684	0.991	0.810
100%	0.827	0.991	0.902
150%	0.914	0.989	0.950
200%	0.955	0.984	0.969
250%	0.975	0.981	0.978
300%	0.981	0.976	0.978
Over+Undersampling			
50%	0.826	0.739	0.780
100%	0.907	0.791	0.845
150%	0.957	0.975	0.966
200%	0.975	0.807	0.883
250%	0.986	0.815	0.892
300%	0.993	0.829	0.903

Table 4.2. Training result of SMOTE-VDM with k=5 and combination with undersampling

Data Sets	Performance		
	R	P	F
SMOTE-VDM k=5			
50%	0.193	0.964	0.322
100%	0.146	0.947	0.252
150%	0.117	0.926	0.208
200%	0.137	0.921	0.239
250%	0.165	0.934	0.280
300%	0.187	0.913	0.310
SMOTE+VDM k=5 and Undersampling			
50%	0.539	0.741	0.624
100%	0.612	0.816	0.700
150%	0.656	0.851	0.741
200%	0.627	0.828	0.714
250%	0.592	0.871	0.705
300%	0.731	0.897	0.805

Table 4.3. Training result of SMOTE-VDM with k=1 and combination with undersampling

Data Sets	Performance		
	R	P	F
SMOTE+VDM k=1			
50%	0.211	0.967	0.346
100%	0.438	0.992	0.608
150%	0.729	0.965	0.831
200%	0.663	0.977	0.790
250%	0.822	0.957	0.884
300%	0.846	0.958	0.898
SMOTE+VDM k=1 and Undersampling			
50%	0.798	0.790	0.794
100%	0.881	0.773	0.823

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

150%	0.895	0.835	0.864
200%	0.962	0.731	0.831
250%	0.934	0.823	0.875
300%	0.963	0.774	0.858

Table 4.4. Prediction test result with same batch test data

Data Sets	Performance		
	R	P	F
Original data set	0.920	1.000	0.958
Undersampling 60%	0.960	1.000	0.980
Oversampling 300%	1.000	1.000	1.000
Over+Undersampling 300%	1.000	0.481	0.649
SMOTE+VDM k=1 300%	0.880	1.000	0.936
SMOTE+VDM k=1 Undersampling 250%	1.000	0.532	0.694
SMOTE+VDM k=5 50%	0.840	1.000	0.913
SMOTE+VDM k=5 Undersampling 300%	0.960	0.686	0.800

Table 4.5. Prediction test result with difference batch test data

Data Sets	Performance		
	R	P	F
Original data set	0.000	0.000	0.000
Undersampling 60%	0.000	0.000	0.000
Oversampling 300%	0.000	0.000	0.000
Over+Undersampling 300%	0.571	0.281	0.376
SMOTE+VDM k=1 300%	0.071	1.000	0.133
SMOTE+VDM k=1 Undersampling 250%	0.500	0.286	0.364
SMOTE+VDM k=5 50%	0.000	0.000	0.000
SMOTE+VDM k=5 Undersampling 300%	0.536	0.333	0.411

prediction test with the testing data from the same batch with training data. The result shows that oversampling perform the best result while the combination type of sampling method did not really perform better compare with other single type of sampling method.

Table 4.6. Prediction test result with another difference batch test data

Data Sets	Performance		
	R	P	F
Original data set	0.000	0.000	0.000
Undersampling 60%	0.000	0.000	0.000
Oversampling 300%	0.000	0.000	0.000
Over+Undersampling 300%	0.346	0.180	0.237
SMOTE+VDM k=1 300%	0.000	0.000	0.000
SMOTE+VDM k=1 Undersampling 250%	0.385	0.196	0.260
SMOTE-VDM k=5 50%	0.000	0.000	0.000
SMOTE+VDM k=5 Undersampling 300%	0.346	0.220	0.269

Both Table 4.5 and 4.6 result outcome were tested with data from different batch. From the result, indicates that sampling method that uses combination with oversampling and undersampling were capable of performing the prediction compare to the single type sampling method which were totally failed. SMOTE+VDM and undersampling method shows better result than plain over + undersampling method.

5. Conclusion

From the training result, undersampling alone was not giving any significant improvement while oversampling method producing the best performance. Oversampling result outperformed our proposed SMOTE-VDM and SMOTE-VDM + undersampling.

However from the prediction test result indicates that, the combination undersampling and oversampling capable of handling wider range of data sets. SMOTE+VDM and undersampling produce robust classifier performance capable of handling better with all those 3 different batches of prediction test data.

From the result analysis, we can see that an exact balance of minority and majority classes are not the main concern to handle the imbalance data sets. The most important matter to focus is the balance distribution of relevant information carried by each class instances. Well balanced number of instances in the data set will produce robust classifier but further improvement on the performance is required.

The result analysis also shows that random undersampling has the potential of information loss which affecting the class precision, whilst oversampling method will improve the class recall with mild impact to the precision but carry the risk of overfitting.

Hence, we conclude that oversampling with appropriate synthetic minority instance is important to improve the class recall with minimum impact to overfitting. As VDM distance function not really suitable with our polynominal data set, distance functions such as entropy based distance function should be considered. On the other hand, because undersampling will cause the information loss and reducing the class precision, different approach on the majority class sampling is required for our future study.

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

6. References

[1] Megat Norulazmi Megat Mohamed Noor, Shaidah Jusoh, *Visualizing the Yield Pattern Outcome for Automatic Data Exploration, ams*, pp. 404-409, 2008 Second Asia International Conference on Modelling & Simulation, 2008.

[2] Peter, W. Tse. Maintenance practices in Hong Kong and the use of the intelligent scheduler. *Journal of Quality in Maintenance Engineering*, 8(4), 369-380, 2002

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O.Hall, W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002) 321–357 Submitted 09/01; published 06/02 AI Access Foundation and Morgan Kaufmann Publishers.

[4] Kihoon Yoon & Stephen Kwek. *A data reduction approach for resolving the imbalanced data issue in functional genomics Journal of Neural Computing & Applications ISSN 0941-0643 (Print) 1433-3058 (Online) Volume 16, Number 3 / May, 2007 Pages 295-306* Springer London.

[5] M. G. Karagiannopoulos, D. S. Anyfantis, S. B. Kotsiantis and P. E. Pintelas. *Local Cost Sensitive Learning for Handling Imbalanced Data Sets Control & Automation, 2007. MED '07. Mediterranean Conference on 27-29 June 2007 page(s): 1-6* Athens, Greece, SBN: 978-1-4244-1282-2

[6] Terry R. Payne and Peter Edwards. *Implicit Feature selection with the value difference metric, Proceedings of the 13th European Conference on Artificial Intelligence, ECAI-98,* John Wiley & Sons, New York, NY, 1998, pp. 450-454.

[7] Ajay D. Joshi. Applying the Wrapper Approach for Auto Discovery of Under-Sampling and Over-Sampling Percentages on Skewed Datasets. A master thesis. 2004

[8] Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco, CA, USA. 1999.

[9] Mierswa, Ingo and Wurst, Michael and Klinkenberg, Ralf and Scholz, Martin and Euler, Timm. *YALE: Rapid Prototyping for Complex Data Mining Tasks, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.*

[10] Batista, G. E. A. P. A., Bazan, A. L., and Monard, M. C. *Balancing Training Data for Automated Annotation of Keywords: a Case Study Journal: Brazilian Workshop on Bioinformatics In WOB (2003), pp. 35—43* of Science in Computer Science. Department of Computer Science and Engineering. College of Engineering University of South Florida. Retrieved on 13th October 2006, from ETD.FCLA website: http://etd.fcla.edu/SF/SFE0000491/Thesis-AjayJoshi.pdf

978-1-4244-2328-6/08/$25.00 © 2008 IEEE