University Library



A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Khorshidi, HA;Aickelin, U

Title:
Constructing classifiers for imbalanced data using diversity optimisation

Date:
2021-07-01

Citation:
Khorshidi, H. A. & Aickelin, U. (2021). Constructing classifiers for imbalanced data using diversity optimisation. Information Sciences, 565, pp.1-16. https://doi.org/10.1016/j.ins.2021.02.069.

Persistent Link:
https://hdl.handle.net/11343/268161

Constructing Classifiers for Imbalanced Data using Diversity Optimisation

Hadi A. Khorshidi[1][0000-0002-2653-4102] and Uwe Aicklin[1][0000-0002-2679-2275]

[1] The University of Melbourne, Melbourne VIC 3010, Australia
{hadi.khorshidi, uwe.aickelin}@unimelb.edu.au

Abstract. Imbalanced data is one of the main challenges for classification models. This paper proposes a new approach for addressing the treatment of imbalanced data using diversity optimisation. For the first time, a novel approach of oversampling the minority group adopts diversity optimisation to generate synthetic instances. Diversity optimisation assures that the generated synthetic instances are close enough to the minority group but not identical. It also ensures the optimal spread of the generated instances in the space. We develop two formulations named as Diversity-based Average Distance Over-sampling (DADO) and Diversity-based Point Wise Over-sampling (DPWO). To evaluate the performance of the proposed formulations, we design experiments using both synthetic data sets and real data sets with unbalanced classes. We examine the performance of the proposed formulations through F1-score and area under curve (AUC) measures in comparison with existing approaches. For this examination, we implement six commonly used classifiers. Our results show that both proposed formulations have potentials to improve the performance of classifiers and DPWO outperforms other comparable re-sampling approaches. We also conduct a sensitivity analysis to investigate the robustness of the formulations and find roadmaps for improvement.

Keywords: Imbalanced Data, Diversity Optimisation, Classification.

1 Introduction

Imbalanced data refer to situations that the frequency of classes is not equally distributed. It is one of the main challenging circumstances in classification algorithms [1]. Classifiers in machine learning are to minimise the misclassification error (or maximise the predictive accuracy). Therefore, the classification algorithms work effectively assuming the number of instances for in classes is approximately balanced. The accuracy of these algorithms would be biased when at least one class has substantially different number of instances. The classification algorithms result in high False Negative rates (FNR) when instances from positive classes are predicted negative [2, 3].

The issue of imbalanced data in machine learning has generally been addressed by two approaches. In the first approach, the learning algorithm of the classifiers is adjusted based on the class with the fewest instances and without changing the training data set. One method to adjust the classification algorithms is the cost-sensitive

learning. In cost-sensitive learning, a variety of costs are assigned to misclassification errors across classes and the classifier does not minimise just a simple misclassification error. The idea is that the misclassification from one class results in higher undesirable outcomes. For example, classifying a person who suffers from cancer as a healthy person is more critical than classifying a healthy person as a person with cancer [4, 5]. Thresholding is a method that adjust the decision boundary for classification [6]. In binary classifiers that estimate probability, the decision boundary is usually 0.5 [7]. Another method to adjust the algorithms is using an ensemble of classifiers to reduce the variance and create a more robust classifier [8].

The second approach, which is the focus of this study, re-samples the training data set to equalise the number of instances in classes. Under-sampling methods are methods that eliminate instances from the class with the majority number of instances. Random under-sampling is the most commonly used under-sampling method. The main drawback of under-sampling is the loss of valuable information from data sets [9]. Another method is over-sampling that replicates the instances of the minority class to balance the data set. The main drawback of over-sampling is that it may increase the chance of overfitting [10]. Also, the existence of a considerable number of identical instances in a data set is a challenging circumstance for classifiers [1]. Applying a combination of over-sampling and under-sampling has also been used to reduce the problem of unbalanced classes [11].

Over-sampling can be used randomly, in a focused way or using synthetic sampling. Focused over-sampling replicates the instances of the minority class where are close to the boundary between majority and minority classes. Over-sampling with replication results in the creation of smaller and more specific decision regions [10]. This problem is addressed by generating synthetic instances from the minority class. Synthetic Minority Oversampling Technique (SMOTE) [3] generates synthetic instances using nearest neighbors of the same class. So, the synthetic instances are not the replication of original instances and broaden the decision region. Using synthetic instances for over-sampling can reduce the FN rate and stabilise the performance of classifiers [4, 10]. In this paper, we aim to propose a new method to generate synthetic instances for the minority class using diversity optimisation. The proposed method ensures that the generated synthetic instances are close to the instances in the minority class but not identical. It maintains the advantages of synthetic over-sampling to treat imbalanced data. Diversity optimisation also ensures the generated instances are spread optimally in the space and broadens the decision region to improve the classification models.

2 Diversity optimisation

Optimisation models usually aim to obtain the best solution to minimise (or maximise) the objective function. However, diversity optimisation is to find a set of solutions that have acceptable objective values and are optimally diverse. There are several benefits for diversity optimisation. First, it provides a variety of options for decision makers to choose from. Sometimes the best solution is hard to achieve but knowing other solutions with almost similar quality makes decision-making easier. Second, some

abstractions and simplifications are normally considered in developing optimisation models. So, the obtained best solution is not necessarily the optimal solution in real world. Third, having a diverse set of almost optimal solutions provides an opportunity to explore more the case study.

Diversity optimisation is to optimise the objective function and optimise the diversity among the solutions. However, it cannot be considered as a typical multi-objective optimisation problem because diversity is not an independent objective function. It does not aim to find diverse with low-quality solutions. Diversity optimisation is a special type of multi-objective optimisation that can be called as mixed multi-objective problem. First, it finds a set of high-quality solutions. Then, it uses a diversity measure to guarantee that the set includes diverse high-quality solutions. The algorithm of diversity optimisation usually contains three main steps. (1) Optimising the objective function: a standard evolutionary algorithm generates pools of solutions and optimises the objective function. (2) Bound adaptation: a bound is determined based on the last generated pool to guarantee the quality of solutions. (3) Diversity maximisation: new solutions are generated to improve the diversity measure under the constraint of the quality bound. These steps iterate to reach the stopping criteria [12].

Diversity optimisation has attracted interests from researchers in recent years. It has been introduced by Ulrich and Thiele [12] for single-objective optimisation problems. It has also been used for multi-objective problems [13]. Jiang and Yang [14] proposed an evolutionary algorithm based on diversity for multi-objective optimisation problems by prioritising diversity over convergence. Gao et al. [15] used diversity optimisation to construct a diverse set of instances for the Traveling Salesperson problem. Neumann et al. [16] adapted popular indicators in multi-objective optimisation such as hypervolume, inverted generational distance and additive epsilon approximation to diversity optimisation. Diversity optimisation has also been used to create a variety of images that are close to the original image but are different [17]. For the first time, we study the adaptation of diversity optimisation to generate synthetic instances which resemble to instances in minority class but differ. These synthetic instances are used to develop a novel method for over-sampling the minority class to treat imbalanced data problems.

3 Diversity-based over-sampling

This section describes the algorithm used for diversity optimisation, the diversity measure, population selection based on diversity and the adaptation of diversity optimisation for over-sampling.

3.1 Diversity optimisation algorithm

Diversity optimisation comprises three steps. The algorithm we use in this study for diversity optimisation is an extended version of NOAH algorithm [12] and has these three steps as well (as shown in Algorithm 1). The input parameters are population size (n), the number of generations for optimising the objective function (g), the number of instances remains in the population after bound adaptation (r), the percentage that

Algorithm 1: Diversity optimisation algorithm

Input: n, g, r, c, v
Output: A diverse set of instances S

1. S = Null; $b = \infty$; $i = 0$
2. **while** $i < c$ **do**
   /* Optimising the objective function */
3. $P \leftarrow$ Randomly generate a population with n instances
4. **for** g generations **do**
5. $P' \leftarrow$ Generate new n instances from P with objective values better than b
6. $P \leftarrow$ Select n best instances from $P \cup P'$
7. **end for**
   /* Bound adaptation */
8. $P \leftarrow$ Select r best instances from $P \cup S$
9. $b' \leftarrow$ Put the objective value of rth best instance in $P \cup S$
10. **if** $b - b' < v \times b$ **then** $i \leftarrow i + 1$ **else** $i \leftarrow 0$
11. $b \leftarrow b'$
    /* Diversity maximisation */
12. $j = 0$
13. **while** $j < c$ **do**
14. $P^{"} \leftarrow$ Generate new r instances from P with objective values better than b
15. $P^{\therefore} \leftarrow$ Select r best diverse instances from $P^{"} \cup S$
16. **end while**
17. **if** diversity of $P^{\therefore}$ is more than S **then** $S \leftarrow P^{\therefore}$ **else** $j \leftarrow j + 1$
18. **end while**

defines the improvement of bound (v) and c which is the stopping criterion diversity maximisation and the whole algorithm. If the population's diversity does not improve for c generations, the diversity maximisation converges. If the bound does not improve for c generations, whole algorithm stops. This parameter facilitates the convergence of the algorithm when the optimal value of the objective function is unknown. In this situation, NOAH algorithm iterates forever.

Every evolutionary algorithm can be applied to optimise the objective function. We use genetic algorithm (GA) as the most popular evolutionary algorithm. So, we use mutation and crossover to generate new instances by considering the bound value. In minimisation problems such as our study, the bound is an upper bound and its initial value is infinity which decreases through the algorithm.

3.2 Diversity-based selection

We use Solow-Polasky measure [18] as the diversity measure. In this diversity measure, a distance matrix ($M = [m_{ij}]$) is constructed and the summation of elements of the distance matrix's inverse (M^{-1}) is the diversity measure ($D(S)$).

$$D(S) = \sum M^{-1} = \sum_i \sum_j e^{-d(s_i, s_j)} \quad (1)$$

where $d(s_i, s_j)$ denotes the distance between elements of set *S* which are instances. We use Euclidean distance in our study.

The Solow-Polasky measure has three properties that are needed for a diversity measure [19]. These three properties are (1) Monotonicity in varieties: the diversity measure increases by adding an individual element that was not in the set, (2) Twining: the diversity measure remains the same by adding an individual element that was already in the set, (3) Monotonicity in distance: the diversity of set *S* should not be smaller than another set S' if all pairs in *S* are as distant as all pairs in S'.

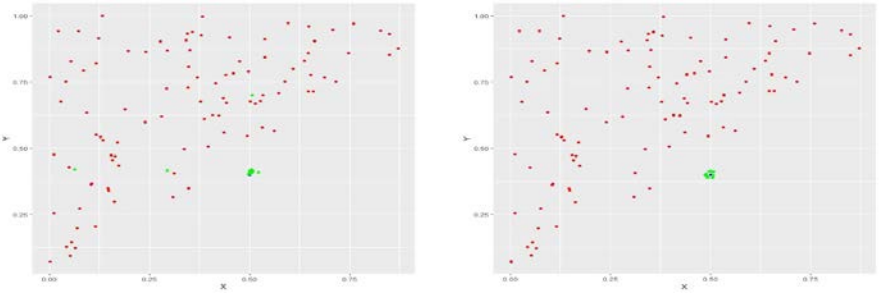
To select the best diverse instances (Algorithm 1, line 15), all possible subsets should be checked. However, this is computationally infeasible. So, we use a greedy approach to discard instances with the least contribution to diversity of the set. The contribution of an instance is defined by the difference between the diversity of the whole set and the diversity of the set without that instance. To reduce the computation, this difference can be formulated as (2) as proved in [12].

$$\sum M^{-1} - \sum A^{-1} = \frac{1}{\bar{c}}(\sum \bar{b} + \bar{c}) \tag{2}$$

where A is the distance matrix of the set without that particular instance, $M = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$, $M^{-1} = \begin{bmatrix} \bar{A} & \bar{b} \\ \bar{b}^T & \bar{c} \end{bmatrix}$, *c* and $\bar{c}$ are single elements, *b* and $\bar{b}$ are vectors and b^T and $\bar{b}^T$ are their transpose.

3.3 Objective functions

To apply the diversity optimisation for over-sampling, a proper objective function is to be defined. As we aim to generate synthetic instances that are close to instances in minority class, we define the objective function as the distance from the instances in the minority class. Therefore, our optimisation problem is a minimisation one. We use Euclidean distance as the distance measure. To clarify the problem, we visualise an example in Fig. 1. The example shows a data set with two features (X and Y) where 99 instances are in class 0 and one instance is in class 1 (minority class).



(a) (b)

Fig. 1. Synthetic instance generation (red, blue and green dots denote class 0, 1 and synthetic instances respectively)

Fig. 1(a) shows the generated instances in the last generation of GA. Some of the instances generated by GA located in the space of the majority class and close instances are clustered together. Whereas, the instances generated by diversity optimisation (Fig. 1(b)) are located close and diversely around the instance in the minor class.

Normally, there are more than one instance in the minority class. To address this situation, we proposed two formulations to implement diversity optimisation. First, we define the objective function as the average of distance from all instances in the minority class. This formulation named as Diversity-based Average Distance Over-sampling (DADO). In DADO, the synthetic instances are diversely spread in the space among the minority class' instances. Second, we divide generating synthetic instances around each instance of the minority class. For example, if we tend to generate 100 synthetic instances and there are 5 instances in the minority class, we generate 20 diverse synthetic instances around each one of those 5. This formulation named as Diversity-based Point Wise Over-sampling (DPWO).

4 Experiments

In this section, we examine the performance of the proposed formulations using both synthetic and real data sets with unbalanced classes. Each data set is randomly divided into training and test data sets by half. We just control to maintain the imbalance ratio across training and test data sets. For example, if the ratio of the majority class to minority one is 95% to 5%, we make sure that both training and test data sets have the same ratio. We develop classifiers on training data set and measure the performance on the test data set. The classifiers are Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT), k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF). We choose NB, KNN and RF as they are sensitive to imbalanced data based on their model assumptions [1]. LR is a commonly used classifier in medical problems even if the data is imbalanced [20]. It also is an effective classifier when classes are linearly separable. DT works based on developing decision regions which are influenced by re-sampling methods [10]. SVM with radial kernel is effective to classify classes which are non-linearly separable.

We measure the performance of the classifiers on test data using F1-score and area under curve (AUC) as classification accuracy is not an appropriate measure for imbalanced data. To calculate F1-score, we need to measure recall and precision which are calculated using (3) and (4). Recall is the proportion of correctly predicted positive instances to all instances in the positive class. Precision is the proportion of correctly predicted positive instances to all predicted positive instances.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

where TP stands for True Positive that is the number of instances from positive class predicted correctly, FN stands for False Negative that is the number of instances from positive classes predicted negative and FP stands for False Positive which is the number

of instances from negative classes predicted positive. F1-score is the harmonic average of recall and precision as (5) [2, 21].

$$Precision = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5}$$

The Receiver Operating Characteristic (ROC) curve is a technique to summarise the performance of a classifier over trade-offs between sensitivity (is the same as recall) and False Positive rate (FPR) as (6).

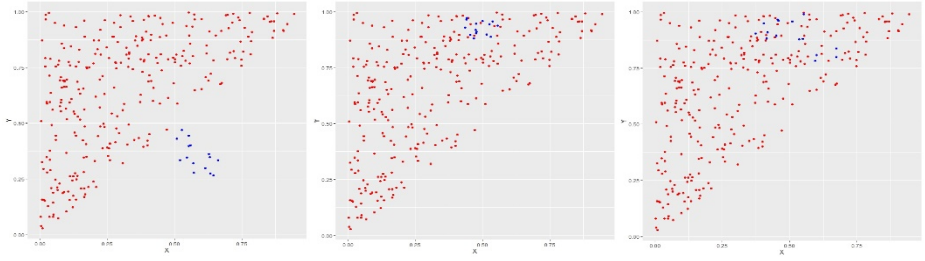
$$FPR = \frac{FP}{TN + FP} \tag{6}$$

where TN stands for True Negative that is the number of instances from negative class predicted correctly. AUC is the area under the ROC curve, is a suitable measure for classifiers' performance especially in the situation of imbalanced data and is independent of the decision boundary [3, 22].

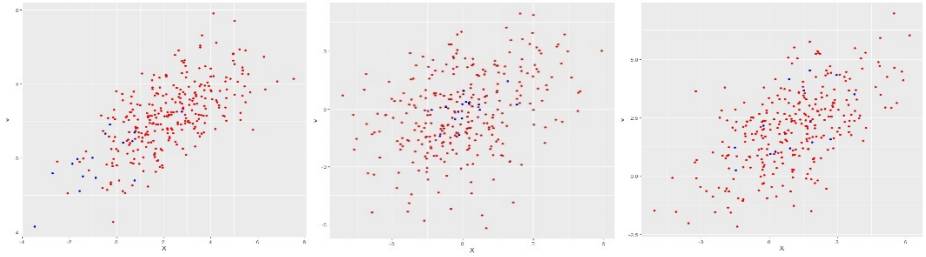
We compare the performance of the proposed formulations with other comparable re-sampling methods such as random over-sampling, random under-sampling, a hybrid of these two and SMOTE [23] method. we run all these methods 30 times and record the values of F1-score and AUC measures. We use mean and Mann-Whitney test, which is a non-parametric statistical test, to compare methods. We also record the measures for original training data set without re-sampling and use 1-sample Wilcoxon test to compare all methods with original measures.

4.1 Synthetic data sets

We create 6 two-dimensional data sets with imbalanced ratio of 95% to 5%. We attempt to cover a variety of situations in these data sets. Fig. 2 visualises these data sets. Data set 1 shows two classes that are linearly separable. In data set 2, two classes are not separable and instances in the minority class have low variance. In data set 3, two classes are not separable as well but the minority class' instances have higher variance. For the remaining data sets, we use multivariate normal distribution to generate feature values. In data set 4, the features in two classes have similar variance and covariance values but different means. In data set 5, the features in two classes have similar means but different variance and covariance values. The minority class is located inside the majority class. In data set 6, first we generate all instances, then we label them randomly. So, there is no specific difference between two classes making it harder to classify.



Data set 1 Data set 2 Data set 3



Data set 4 Data set 5 Data set 6

Fig. 2. Synthetic data sets

Table 1 presents the outcomes of experiments on these data sets. The best value in average for each row is indicated as a bold number. The results of non-parametric tests at 1% significant level are presented as well. For example, 5+ means the measure values for that method are significantly better than method (5), which is under-sampling. It worth to note that NA value for F1-score means TP is equal to zero. The proposed formulations show great potential to handle imbalanced data and outperform other re-sampling methods. Especially in data set 6, which is a hard data set, DPWO shows a great performance. Another interesting observation is that re-sampling methods show various performances in different cases. In some cases, the original training data set results in better performances. It shows that dealing with imbalanced data is the research area that still needs attention from the machine learning community.

4.2 Real data sets

We conduct the experiments on three real data sets that cover different ranges imbalanced ratios. The details of these data sets are brought in Table 2. Hepatitis and Pima data sets are from UCI repository [24]. Oil data set has been used to detect oil spills from satellite radar images [25] and is accessible from [26]. Table 3 presents the outcomes of the experiments on real data sets. The proposed formulations show great potential in real data sets compared with other comparable methods. Still, the original training data set results in better performances in few cases.

Table 2. Data sets description.

Data set	# of instances	# of features	Missing	Imbalanced ratio
Pima	768	8	No	65% - 35%
Hepatitis	155	19	Yes	80% - 20%
Oil	937	49	No	95% - 5%

Table 1. Results for synthetic data sets.

Data set	Measure	Method	Original (1)	DPWO (2)		DADO (3)		Over-sampling (4)		Under-sampling (5)		Hybrid (6)		SMOTE (7)	
				Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat
Data set 1	F1	LR	**1**	**1**	5^{+}	**1**	5^{+}	**1**	5^{+}	0.61		**1**	5^{+}	**1**	5^{+}
		NB	0.5	0.909	$1^{+}3^{+}7^{+}$	NA		**0.961**	$1^{+}2^{+}3^{+}7^{+}$	0.955	$1^{+}2^{+}3^{+}7^{+}$	0.952	$1^{+}2^{+}3^{+}7^{+}$	0.855	$1^{+}3^{+}$
		DT	NA	**1**	$1^{+}3^{+}5^{+}6^{+}$	0.8	$1^{+}5^{+}6^{+}$	**1**	$1^{+}3^{+}5^{+}6^{+}$	NA		0.661	$1^{+}5^{+}$	**1**	$1^{+}3^{+}5^{+}6^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	**1**	**1**	5^{+}	**1**	5^{+}	**1**	5^{+}	0.852		**1**	5^{+}	**1**	5^{+}
		RF	0.667	0.789	$1^{+}5^{+}$	0.796	$1^{+}5^{+}$	0.778	$1^{+}5^{+}$	0.45		**0.867**	$1^{+}2^{+}3^{+}4^{+}5^{+}$	0.829	$1^{+}2^{+}3^{+}4^{+}5^{+}$
	AUC	LR	**1**	**1**	5^{+}	**1**	5^{+}	**1**	5^{+}	0.973		**1**	5^{+}	**1**	5^{+}
		NB	**1**	**1**	3^{+}	0.701		**1**	3^{+}	0.997	3^{+}	**1**	3^{+}	**1**	3^{+}
		DT	0.5	**1**	$1^{+}3^{+}5^{+}6^{+}$	0.858	$1^{+}5^{+}$	**1**	$1^{+}3^{+}5^{+}6^{+}$	0.5		0.905	$1^{+}3^{+}5^{+}$	**1**	$1^{+}3^{+}5^{+}6^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		RF	**1**	**1**		**1**		**1**		0.999		**1**		**1**	
Data set 2	F1	LR	NA	**0.276**	$1^{+}3^{+}4^{+}6^{+}7^{+}$	0.241	$1^{+}4^{+}7^{+}$	0.224	1^{+}	0.272	$1^{+}3^{+}4^{+}6^{+}7^{+}$	0.246	$1^{+}4^{+}7^{+}$	0.228	1^{+}
		NB	0.4	0.475	$1^{+}3^{+}7^{+}$	0.298		0.475	$1^{+}3^{+}7^{+}$	**0.508**	$1^{+}2^{+}3^{+}4^{+}7^{+}$	0.491	$1^{+}3^{+}7^{+}$	0.442	$1^{+}3^{+}$
		DT	NA	0.333	$1^{+}3^{+}4^{+}5^{+}6^{+}$	0.237	$1^{+}4^{+}5^{+}$	0.221	$1^{+}5^{+}$	NA		0.23	$1^{+}4^{+}5^{+}$	**0.500**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	NA	0.55	$1^{+}5^{+}6^{+}$	0.557	$1^{+}5^{+}6^{+}$	0.535	$1^{+}5^{+}$	0.378	1^{+}	0.516	$1^{+}5^{+}$	**0.588**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
		RF	NA	0.29	$1^{+}4^{+}$	**0.329**	$1^{+}4^{+}$	0.264	1^{+}	0.317	$1^{+}4^{+}$	0.315	$1^{+}4^{+}$	0.321	$1^{+}4^{+}$
	AUC	LR	**0.88**	**0.88**	5^{+}	**0.88**	5^{+}	**0.88**	5^{+}	0.843		**0.88**	5^{+}	**0.88**	5^{+}
		NB	**0.956**	**0.956**	$3^{+}7^{+}$	0.910		0.955	3^{+}	**0.956**	$3^{+}7^{+}$	**0.956**	3^{+}	0.954	3^{+}
		DT	0.5	**0.728**	$1^{+}5^{+}$	0.601	$1^{+}5^{+}$	0.626	$1^{+}5^{+}$	0.500		0.694	$1^{+}5^{+}$	0.664	$1^{+}5^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	0.964	**0.975**	$1^{+}3^{+}5^{+}$	0.962		0.974	$1^{+}3^{+}5^{+}$	0.965		0.972	$1^{+}3^{+}5^{+}$	**0.975**	$1^{+}3^{+}5^{+}$
		RF	0.922	**0.945**	$1^{+}3^{+}5^{+}6^{+}$	0.931	$1^{+}3^{+}5^{+}$	**0.945**	$1^{+}3^{+}5^{+}6^{+}$	0.908		0.927	$5^{+}6^{+}$	**0.945**	$1^{+}3^{+}5^{+}6^{+}$
Data set 3	F1	LR	NA	0.206	$1^{+}6^{+}$	**0.24**	$1^{+}2^{+}4^{+}5^{+}6^{+}7^{+}$	0.202	1^{+}	0.224	1^{+}	0.199	1^{+}	0.206	$1^{+}6^{+}$
		NB	NA	0.25	$1^{+}3^{+}4^{+}5^{+}$	0.211	1^{+}	0.244	1^{+}	0.219	1^{+}	0.246	1^{+}	**0.265**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
		DT	NA	0.209	$1^{+}5^{+}$	**0.33**	$1^{+}2^{+}4^{+}5^{+}6^{+}7^{+}$	0.223	$1^{+}5^{+}$	NA		0.265	$1^{+}5^{+}$	0.226	$1^{+}5^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	NA	0.432	$1^{+}3^{+}5^{+}6^{+}7^{+}$	0.267	1^{+}	**0.447**	$1^{+}2^{+}3^{+}5^{+}6^{+}7^{+}$	0.237	1^{+}	0.341	$1^{+}3^{+}5^{+}$	0.313	$1^{+}3^{+}5^{+}$
		RF	NA	0.267	$1^{+}5^{+}7^{+}$	0.3	$1^{+}5^{+}7^{+}$	0.286	$1^{+}5^{+}7^{+}$	0.219	1^{+}	**0.311**	$1^{+}5^{+}7^{+}$	0.232	1^{+}
	AUC	LR	0.846	**0.85**	$1^{+}3^{+}5^{+}7^{+}$	0.844	5^{+}	0.849	$1^{+}3^{+}5^{+}7^{+}$	0.804		0.847	$1^{+}3^{+}5^{+}7^{+}$	0.844	5^{+}
		NB	**0.85**	**0.85**	3^{+}	0.804		**0.85**	3^{+}	0.847	3^{+}	**0.85**	3^{+}	**0.85**	3^{+}
		DT	0.5	0.578	$1^{+}5^{+}$	**0.779**	$1^{+}2^{+}4^{+}5^{+}6^{+}7^{+}$	0.593	$1^{+}5^{+}$	0.5		0.668	$1^{+}2^{+}4^{+}5^{+}$	0.646	$1^{+}2^{+}4^{+}5^{+}$
		KNN	**1**	**1**		**1**		**1**		**1**		**1**		**1**	
		SVM	**0.926**	0.916	$3^{+}5^{+}6^{+}7^{+}$	0.728		0.916	$3^{+}5^{+}6^{+}7^{+}$	0.867	3^{+}	0.908	$3^{+}5^{+}$	0.911	$3^{+}5^{+}$
		RF	0.77	0.746	7^{+}	0.774	$1^{+}2^{+}4^{+}7^{+}$	0.732	7^{+}	**0.833**	$1^{+}2^{+}3^{+}4^{+}7^{+}$	0.821	$1^{+}2^{+}3^{+}4^{+}7^{+}$	0.701	
Data set 4	F1	LR	NA	0.159	1^{+}	**0.21**	$1^{+}2^{+}4^{+}5^{+}7^{+}$	0.173	$1^{+}2^{+}7^{+}$	0.166	$1^{+}2^{+}$	0.17	$1^{+}2^{+}7^{+}$	0.162	$1^{+}3^{+}$
		NB	NA	0.169	1^{+}	0.167	1^{+}	**0.172**	1^{+}	0.169	1^{+}	0.166	1^{+}	**0.172**	$1^{+}3^{+}$
		DT	NA	0.102	$1^{+}5^{+}$	**0.188**	$1^{+}2^{+}5^{+}$	0.187	$1^{+}2^{+}5^{+}$	NA		0.157	5^{+}	0.186	$1^{+}2^{+}5^{+}$
		KNN	0.25	**0.914**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.444	$1^{+}5^{+}6^{+}$	0.554	$1^{+}3^{+}5^{+}6^{+}$	0.22		0.383	$1^{+}5^{+}$	0.811	$1^{+}3^{+}4^{+}5^{+}6^{+}$
		SVM	NA	0.044	1^{+}	**0.235**	$1^{+}2^{+}4^{+}5^{+}6^{+}7^{+}$	0.108	$1^{+}2^{+}$	0.149	$1^{+}2^{+}4^{+}7^{+}$	0.142	$1^{+}2^{+}6^{+}4^{+}$	0.144	$1^{+}2^{+}4^{+}$
		RF	0.25	0.189	5^{+}	0.215	$2^{+}5^{+}$	**0.392**	$1^{+}2^{+}3^{+}5^{+}7^{+}$	0.139		0.23	$2^{+}5^{+}$	0.204	5^{+}
	AUC	LR	0.737	**0.747**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.728		0.744	$1^{+}3^{+}5^{+}7^{+}$	0.73		0.742	$1^{+}3^{+}5^{+}$	0.741	$1^{+}3^{+}5^{+}$
		NB	**0.75**	0.684		0.687	2^{+}	**0.75**	$2^{+}3^{+}5^{+}7^{+}$	0.735	$2^{+}3^{+}6^{+}$	**0.75**	$2^{+}3^{+}5^{+}7^{+}$	0.734	$2^{+}3^{+}$
		DT	0.5	0.707	$1^{+}3^{+}4^{+}5^{+}6^{+}$	0.633	$1^{+}5^{+}$	0.584	$1^{+}5^{+}$	0.5		0.655	$1^{+}3^{+}4^{+}5^{+}$	**0.729**	$1^{+}3^{+}4^{+}5^{+}6^{+}$
		KNN	**0.843**	0.303		0.305	2^{+}	0.521	$2^{+}3^{+}6^{+}7^{+}$	0.497	$2^{+}3^{+}7^{+}$	0.495	$2^{+}3^{+}7^{+}$	0.453	$2^{+}3^{+}$
		SVM	0.592	0.504		0.495		0.616	$1^{+}2^{+}3^{+}$	**0.689**	$1^{+}2^{+}3^{+}4^{+}6^{+}7^{+}$	0.638	$1^{+}5^{+}$	0.667	$1^{+}2^{+}3^{+}4^{+}6^{+}$
		RF	0.638	0.648		0.671	$1^{+}4^{+}$	0.64		0.665	$1^{+}2^{+}4^{+}$	0.668	4^{+}	**0.734**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
Data set 5	F1	LR	NA	0.072	$1^{+}7^{+}$	0.093	$1^{+}2^{+}4^{+}6^{+}7^{+}$	0.086	$1^{+}2^{+}7^{+}$	**0.1**	$1^{+}2^{+}4^{+}6^{+}7^{+}$	0.075	$1^{+}7^{+}$	0.036	1^{+}
		NB	NA	0.275	$1^{+}3^{+}$	NA		0.287	$1^{+}2^{+}3^{+}$	0.277	1^{+}	0.29	$1^{+}2^{+}3^{+}5^{+}$	**0.325**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
		DT	NA	0.222	$1^{+}3^{+}5^{+}$	NA		0.285	$1^{+}3^{+}5^{+}$	NA		0.281	$1^{+}3^{+}5^{+}$	**0.287**	$1^{+}2^{+}3^{+}5^{+}$
		KNN	0.923	**1**	$1^{+}3^{+}4^{+}5^{+}6^{+}$	0.923	$5^{+}6^{+}$	0.923	$5^{+}6^{+}$	0.153		0.767	5^{+}	**1**	$1^{+}3^{+}4^{+}5^{+}6^{+}$
		SVM	NA	**0.35**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.235	$1^{+}7^{+}$	0.303	$1^{+}3^{+}5^{+}$	0.243	$1^{+}3^{+}$	0.305	$1^{+}3^{+}5^{+}$	0.337	$1^{+}3^{+}4^{+}5^{+}6^{+}$
		RF	NA	0.181	$1^{+}3^{+}4^{+}$	NA		NA		0.206	$1^{+}3^{+}4^{+}$	0.221	$1^{+}3^{+}4^{+}$	**0.33**	$1^{+}2^{+}3^{+}4^{+}5^{+}6^{+}$
	AUC	LR	0.559	**0.575**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.554	5^{+}	0.559	$1^{+}3^{+}5^{+}7^{+}$	0.534		0.538		0.564	$1^{+}3^{+}5^{+}6^{+}$
		NB	0.885	**0.902**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.817		0.887	$2^{+}7^{+}$	0.888	$3^{+}6^{+}7^{+}$	0.89	$1^{+}3^{+}7^{+}$	0.882	$1^{+}3^{+}$
		DT	0.5	0.723	$1^{+}3^{+}4^{+}5^{+}$	0.343	5^{+}	0.662	$1^{+}5^{+}$	0.5		**0.759**	$1^{+}2^{+}3^{+}4^{+}5^{+}$	0.752	$1^{+}3^{+}4^{+}5^{+}$
		KNN	**1**	**1**	5^{+}	**1**	5^{+}	**1**	5^{+}	0.926		**1**	5^{+}	**1**	5^{+}
		SVM	0.876	**0.923**	$1^{+}3^{+}4^{+}5^{+}6^{+}7^{+}$	0.899	1^{+}	0.919	$1^{+}3^{+}5^{+}$	0.901	$1^{+}7^{+}$	0.916	$1^{+}5^{+}$	0.919	$1^{+}3^{+}5^{+}$

		RF	0.849	**0.91**	$1^+3^+4^+5^+6^+$	0.815		0.873	$1^+3^+5^+$	0.808		0.884	$1^+3^+5^+$	0.907	$1^+3^+4^+5^+6^+$
Data set 6	F1	LR	NA	**0.138**	$1^+4^+5^+6^+7^+$	0.13	$1^+4^+5^+6^+7^+$	0.056	1^+	0.085	$1^+4^+7^+$	0.083	$1^+4^+7^+$	0.057	1^+
		NB	NA	0.079	$1^+4^+6^+7^+$	**0.108**	$1^+2^+4^+6^+7^+$	0.067	1^+	0.105	$1^+2^+4^+6^+7^+$	NA		0.058	1^+
		DT	NA	**0.24**	$1^+3^+4^+5^+6^+7^+$	NA		0.091	1^+	NA		0.074	1^+	0.084	1^+
		KNN	NA	0.643	$1^+3^+4^+5^+$	0.521	$1^+4^+5^+$	0.471	1^+5^+	0.182	1^+	0.663	$1^+3^+4^+5^+$	**0.722**	$1^+2^+3^+4^+5^+6^+$
		SVM	NA	**0.248**	$1^+3^+4^+5^+6^+7^+$	0.148	$1^+4^+5^+6^+7^+$	0.102	1^+7^+	0.106	1^+7^+	0.109	1^+7^+	0.078	1^+
		RF	NA	**0.144**	$1^+3^+4^+5^+6^+7^+$	NA		NA		0.118	$1^+3^+4^+6^+7^+$	NA		0.066	$1^+3^+4^+6^+$
	AUC	LR	0.593	0.598	$1^+3^+4^+5^+$	0.563		0.586		0.583		0.62	$1^+2^+3^+4^+5^+$	**0.641**	$1^+2^+3^+4^+5^+$
		NB	0.57	0.456		0.559	2^+6^+	0.594	$1^+2^+3^+6^+$	0.593	$1^+2^+3^+6^+$	0.489	2^+	**0.669**	$1^+2^+3^+4^+5^+6^+$
		DT	0.5	**0.797**	$1^+3^+4^+5^+6^+7^+$	0.632	$1^+4^+5^+6^+7^+$	0.466		0.5	4^+6^+	0.443		0.544	$1^+4^+5^+6^+$
		KNN	0.735	**0.8**	$1^+3^+4^+5^+$	0.739	1^+5^+	0.741	$1^+3^+5^+$	0.664		**0.8**	$1^+3^+4^+5^+$	**0.8**	$1^+3^+4^+5^+$
		SVM	0.486	**0.586**	$1^+3^+4^+5^+6^+7^+$	0.539	$1^+4^+5^+6^+7^+$	0.534	$1^+5^+6^+7^+$	0.567		0.571	1^+5^+	0.551	$1^+5^+6^+$
		RF	0.651	**0.726**	$1^+3^+4^+5^+6^+7^+$	0.525		0.633	$3^+5^+6^+7^+$	0.583	3^+6^+	0.52		0.566	

Table 3. Results for real data sets.

Data set	Meas-ure	Method	Origi-nal (1)	DPWO (2)		DADO (3)		Over-sampling (4)		Under-sampling (5)		Hybrid (6)		SMOTE (7)	
				Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat	Mean	Stat
Pima	F1	LR	0.646	0.653	1^+	0.649		0.685	$1^+2^+3^+6^+$	0.684	$1^+2^+3^+6^+$	0.667	1^+3^+	**0.692**	$1^+2^+3^+5^+6^+$
		NB	0.657	0.63	3^+	0.593		**0.672**	$1^+2^+3^+6^+$	0.663	2^+3^+	0.652	2^+3^+	0.671	$1^+2^+3^+6^+$
		DT	0.583	0.609	1^+	0.631	$1^+2^+6^+$	0.645	$1^+2^+6^+$	0.647	$1^+2^+6^+$	0.601		**0.65**	$1^+2^+6^+$
		KNN	0.557	0.624	$1^+3^+6^+$	0.547	6^+	0.645	$1^+3^+6^+$	**0.655**	$1^+2^+3^+6^+$	0.257		0.645	$1^+2^+3^+6^+$
		SVM	0.644	**0.69**	$1^+3^+4^+6^+$	0.654	1^+	0.676	1^+3^+	0.684	$1^+3^+6^+$	0.659	1^+	0.688	$1^+3^+6^+$
		RF	0.626	0.659	1^+	0.661	1^+6^+	0.671	1^+6^+	**0.687**	$1^+2^+3^+6^+$	0.654	1^+	0.671	1^+6^+
	AUC	LR	**0.832**	0.819	6^+	0.827	$1^+2^+4^+5^+$	0.828	2^+6^+	0.825	6^+	0.806		0.831	$2^+3^+5^+6^+$
		NB	0.812	0.797	3^+6^+	0.754		0.811	$2^+3^+6^+$	0.803	3^+6^+	0.785	3^+	**0.816**	$1^+2^+3^+5^+6^+$
		DT	0.732	0.729	6^+	**0.766**	$1^+2^+4^+6^+$	0.753	$1^+2^+6^+$	0.762	$1^+2^+6^+$	0.718		0.755	$1^+2^+6^+$
		KNN	0.783	**0.79**	$1^+3^+4^+5^+6^+7^+$	0.774	$4^+6^+7^+$	0.761	6^+	0.778	$4^+6^+7^+$	0.745		0.764	6^+
		SVM	0.828	0.827	6^+	**0.83**	$1^+2^+4^+5^+6^+7^+$	0.824	6^+	0.825	6^+	0.795		0.826	6^+
		RF	0.816	0.812	6^+	**0.824**	$1^+2^+4^+5^+6^+7^+$	0.82	$1^+2^+6^+$	0.815	6^+	0.793		0.816	6^+
Hepatitis	F1	LR	0.235	**0.26**	$1^+3^+7^+$	0.204		0.239	$1^+3^+7^+$	**0.26**	$1^+3^+7^+$	**0.26**	$1^+3^+7^+$	0.235	3^+
		NB	**0.593**	0.454	3^+	0.387		0.542	$1^+2^+3^+5^+$	0.492	3^+	0.522	$1^+2^+3^+5^+$	0.579	$1^+2^+3^+4^+5^+6^+$
		DT	0.3	**0.394**	$1^+3^+4^+5^+$	0.256		0.328	$1^+3^+5^+$	NA		0.378	$1^+3^+4^+5^+$	0.379	$1^+3^+4^+5^+$
		KNN	NA	**0.364**	$1^+3^+4^+5^+6^+7^+$	0.167		0.313	$1^+3^+5^+6^+7^+$	0.167		0.257	$1^+3^+5^+7^+$	0.196	
		SVM	0.461	0.456	5^+	**0.462**	1^+5^+	0.457	5^+	0.389	5^+	0.461	5^+	0.459	5^+
		RF	0.4	0.392	4^+7^+	0.377	4^+7^+	0.352		**0.457**	$3^+4^+6^+7^+$	0.373	4^+7^+	0.349	
	AUC	LR	0.495	0.544	$1^+3^+4^+7^+$	0.486	7^+	0.479	7^+	0.572	$1^+3^+4^+7^+$	**0.576**	$1^+3^+4^+7^+$	0.477	
		NB	0.859	0.841	5^+	**0.872**	$1^+2^+4^+5^+6^+$	0.853	5^+	0.812		0.842	5^+	0.866	$1^+2^+5^+6^+$
		DT	0.574	0.621	$1^+3^+4^+5^+$	0.56	5^+	0.614	$1^+3^+5^+$	0.500		0.648	$1^+3^+4^+5^+$	**0.662**	$1^+3^+4^+5^+$
		KNN	0.533	**0.575**	$1^+3^+5^+6^+7^+$	0.551	$1^+5^+7^+$	0.561	$1^+5^+7^+$	0.448		0.536	5^+	0.508	5^+
		SVM	0.741	0.813	$1^+4^+5^+$	**0.83**	$1^+2^+4^+5^+7^+$	0.782	1^+5^+	0.716		0.806	1^+5^+	0.801	$1^+4^+5^+$
		RF	0.777	0.779		0.774		0.784		0.779		0.783		**0.792**	1^+3^+
Oil	F1	LR	**0.439**	0.392	5^+	0.407	$4^+5^+6^+7^+$	0.374	5^+	0.107		0.354	5^+	0.387	5^+
		NB	0.069	NA		NA		0.066	2^+3^+	**0.148**	$1^+2^+3^+4^+7^+$	0.091	$1^+2^+3^+4^+7^+$	0.062	2^+3^+
		DT	0.286	0.262	6^+	0.306	$1^+2^+5^+6^+$	0.313	$1^+2^+5^+6^+$	0.25	6^+	0.197		**0.328**	$1^+2^+3^+5^+6^+$
		KNN	NA	**0.277**	$1^+3^+4^+5^+6^+7^+$	0.083	1^+	0.246	$1^+3^+5^+$	0.149	1^+3^+	0.257	$1^+3^+4^+5^+$	0.255	$1^+3^+5^+$
		SVM	NA	NA		NA		NA		NA		NA		NA	
		RF	0.091	0.29	$1^+4^+5^+$	0.282	$1^+4^+5^+$	0.202	1^+	0.238	1^+	**0.435**	$1^+2^+3^+4^+5^+$	0.432	$1^+2^+3^+4^+5^+$
	AUC	LR	0.727	0.714	4^+5^+	**0.735**	$1^+2^+4^+5^+$	0.688	5^+	0.602		0.721	4^+5^+	0.722	4^+5^+
		NB	0.51	0.566	$1^+3^+4^+7^+$	0.506		0.517		**0.664**	$1^+3^+4^+7^+$	0.548	$1^+3^+7^+$	0.492	
		DT	0.564	0.694	1^+3^+	0.622	1^+	0.671	1^+3^+	**0.755**	$1^+2^+3^+4^+6^+7^+$	0.708	$1^+3^+4^+$	0.735	$1^+2^+3^+4^+6^+$
		KNN	0.652	0.682	$1^+3^+4^+$	0.652	4^+	0.649		0.682	3^+4^+	0.671	$1^+3^+4^+$	**0.797**	$1^+2^+3^+4^+5^+6^+$
		SVM	0.5	0.5		0.5		0.5		0.5		0.5		0.5	
		RF	0.846	**0.852**	$1^+3^+5^+$	0.831	5^+	0.85	3^+5^+	0.793		**0.852**	$1^+3^+5^+$	0.85	3^+5^+

As Tables 1 and 3 are large and difficult to interpret, we summarise those results in Table 4. Across all data sets and classifiers, for each method, we record the number of cases it has resulted in the best value in average (#best), the average of the number of other methods it has significantly outperformed (#OP), and the percentage of the cases it has been significantly better than the original (%(1)).

Table 4. Summary of results.

Meas-ure	DPWO			DADO			Over-sampling			Under-sampling			Hybrid			SMOTE		
	#best	#OP	%(1)	#best	#OP	%(1)	#best	#OP	%(1)	#best	#OP	%(1)	#best	#OP	%(1)	#best	#OP	%(1)
F1	**19**	**2.75**	**85.2**	13	1.94	64.8	11	2.19	75.9	10	1.72	55.5	9	2.11	66.7	18	**2.75**	75.9
AUC	**28**	**2.57**	**50**	15	1.65	37	13	1.93	40.5	9	1.2	16.7	16	1.85	42.6	21	2.46	**50**

Table 4 shows that DPWO performs the best among the re-sampling methods considering all three metrics in terms of F1 and AUC. DPWO significantly improves the F1-score of original classifiers in more than 85% cases. SMOTE has the second highest performance. DADO has the third rank based on #best and improves the F1-score of original classifiers significantly in about 65% cases.

DPWO performs the best in overall because it generates synthetic instances diversely around each original instance. So that the chance of having instances similar to instances of minority class in test data (unknown future) get higher and classifiers can develop more generalised classification rules. However, DADO generates the synthetic instances in the space between original instances. It may result in generating instances in the space that does not belong to the minority class and misleading the classifiers. It does not mean that DADO always misleads the classifiers. DADO, by generating instances in the space between original instances, highlights the space of the minority class. It helps the classifiers like DT and SVM to determine the decision region better to classify minority instances more correctly like as shown in the results for data set 4.

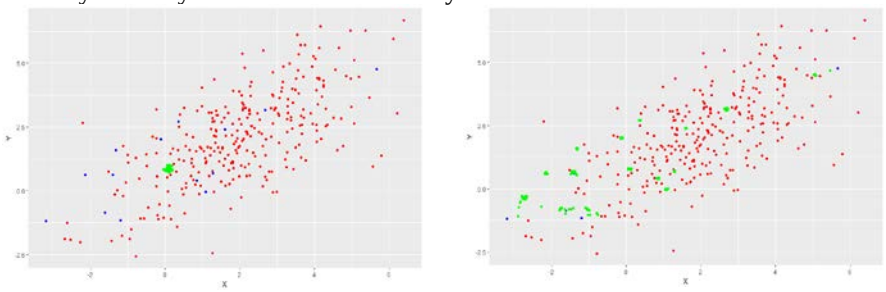


Fig. 3. Synthetic data generation for data set 4

Fig. 3 shows how DADO and DPWO have performed for data set 4. DADO generates synthetic instances in a space that is helpful for DT and SVM to determine the best decision region for classification. DPWO generates synthetic instances around the original instances where the instances from test data would be there. These synthetic instances are helpful for KNN classifier.

4.3 Sensitivity analysis

Both objective function and diversity measure are formulated using a distance measure which we have used the Euclidean distance. In this section, we investigate how

changing the distance measure can impact the performance of DPWO and DADO. So, we replace the Euclidean distance with the summation of absolute differences of elements, which is called Manhattan distance [27, 28]. We repeat the experiments on real data sets using the Manhattan distance and examine whether F1-score and AUC are different significantly using Mann-Whitney test. Table 5 summarises the results for each formulation that shows for how many cases there is no significant difference, and for how many cases either of distance measures result in better performance.

Table 5. Sensitivity analysis results.

Formulation	Euclidean	No Difference	Manhattan
DPWO	10	26	0
DADO	3	22	11

In most cases (more than 65%), there is no significant differences. This shows that both formulations are almost robust in terms of changing distance measure from Euclidean to Manhattan. However, by investigating those cases that have significant difference, DPWO has better results using Euclidean while DADO has better results using Manhattan. It concludes that using Manhattan distance can improve the performance of DADO.

5 Conclusions

In this paper, we have introduced the novel application of diversity optimisation for over-sampling through generating synthetic instances. We have proposed two formulations as DPWO and DADO to adapt diversity optimisation in constructing classifiers for imbalanced data by generating diverse synthetic instances close to the instances in the minority class. We have examined the performance of the proposed formulations through an extensive experimental design using both synthetic and real data, six classifiers and appropriate measures in comparison with existing re-sampling methods. The results have shown both formulations are powerful to improve the performance of the classifiers for imbalanced data, and DPWO outperforms other comparable methods.

This study is an initial work in using diversity optimisation for over-sampling and shows great potentials. So, further research can be conducted to improve the current study. Moreover, the results have shown that there is a need to explore new re-sampling methods to deal with imbalanced data, as the existing methods do not have stable performance. According to sensitivity analysis, a suggestion for further research would be to investigate the impact of distance measures to improve the performance. Other suggestions are developing new formulations, using various diversity measures and proposing hybrid approaches.

References

1. Muñoz, M.A., Villanova, L., Baatar, D., Smith-Miles, K.: Instance spaces for machine learning classification. Machine Learning 107, 109-147 (2018)

2. Thabtah, F., Hammoud, S., Kamalov, F., Gonsalves, A.: Data imbalance in classification: Experimental evaluation. Information Sciences 513, 429-441 (2020)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321-357 (2002)
4. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis 6, 429-449 (2002)
5. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 155-164. (1999)
6. Zou, Q., Xie, S., Lin, Z., Wu, M., Ju, Y.: Finding the Best Classification Threshold in Imbalanced Classification. Big Data Research 5, 2-8 (2016)
7. Radwan, A.M.: Enhancing prediction on imbalance data by thresholding technique with noise filtering. In: 2017 8th International Conference on Information Technology (ICIT), pp. 399-404. (2017)
8. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence 23, 687-719 (2009)
9. Guo, X., Yin, Y., Dong, C., Yang, G., Zhou, G.: On the Class Imbalance Problem. In: 2008 Fourth International Conference on Natural Computation, pp. 192-201. (2008)
10.Chawla, N.V.: Data Mining for Imbalanced Datasets: An Overview. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 875-886. Springer US, Boston, MA (2010)
11.Akbarzadeh Khorshidi, H., Hassani-Mahmooei, B., Haffari, G.: An Interpretable Algorithm on Post-injury Health Service Utilization Patterns to Predict Injury Outcomes. Journal of Occupational Rehabilitation (2019)
12.Ulrich, T., Thiele, L.: Maximizing population diversity in single-objective optimization. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, pp. 641-648. (2011)
13.Ulrich, T., Bader, J., Zitzler, E.: Integrating decision space diversity into hypervolume-based multiobjective search. Proceedings of the 12th annual conference on Genetic and evolutionary computation, pp. 455–462. Association for Computing Machinery, Portland, Oregon, USA (2010)
14.Jiang, S., Yang, S.: Convergence Versus Diversity in Multiobjective Optimization. In: Parallel Problem Solving from Nature – PPSN XIV, pp. 984-993. Springer International Publishing, (2016)
15.Gao, W., Nallaperuma, S., Neumann, F.: Feature-Based Diversity Optimization for Problem Instance Classification. In: Parallel Problem Solving from Nature – PPSN XIV, pp. 869-879. Springer International Publishing, (2016)
16.Neumann, A., Gao, W., Wagner, M., Neumann, F.: Evolutionary diversity optimization using multi-objective indicators. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 837-845. (2019)
17.Alexander, B., Kortman, J., Neumann, A.: Evolution of artistic image variants through feature based diversity optimisation. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 171-178. (2017)
18.Solow, A.R., Polasky, S.: Measuring biological diversity. Environmental and Ecological Statistics 1, 95-103 (1994)

19.Ulrich, T., Bader, J., Thiele, L.: Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search. In: Parallel Problem Solving from Nature, PPSN XI, pp. 707-717. Springer Berlin Heidelberg, (2010)
20.Protopapa, K.L., Simpson, J.C., Smith, N.C.E., Moonesinghe, S.R.: Development and validation of the Surgical Outcome Risk Tool (SORT). Br J Surg 101, 1774-1783 (2014)
21.Sasaki, Y.: The truth of the F-measure. https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf
22.Wong, D.J.N., Oliver, C.M., Moonesinghe, S.R.: Predicting postoperative morbidity in adult elective surgical patients using the Surgical Outcome Risk Tool (SORT). BJA: British Journal of Anaesthesia 119, 95-105 (2017)
23.Siriseriwan, W.: smotefamily: A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE. https://cran.r-project.org/web/packages/smotefamily/index.html
24.Dua, D., Karra Taniskidou, E.: UCI machine learning repository. University of California, School of Information and Computer Science,
25.Kubat, M., Holte, R.C., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning 30, 195-215 (1998)
26.Brownlee, J.: How to Develop an Imbalanced Classification Model to Detect Oil Spills. https://machinelearningmastery.com/imbalanced-classification-model-to-detect-oil-spills/
27.Huang, J., Mao, B., Bai, Y., Zhang, T., Miao, C.: An Integrated Fuzzy C-Means Method for Missing Data Imputation Using Taxi GPS Data. Sensors 20, 1992 (2020)
28.Akbarzadeh Khorshidi, H., Aickelin, U., Haffari, G., Hassani-Mahmooei, B.: Multi-objective semi-supervised clustering to identify health service patterns for injured patients. Health Information Science and Systems 7, 18 (2019)