A Swarm Intelligence Approach in Undersampling Majority Class

Haya Abdullah Alhakbani* and Mohammad Majid al-Rifaie

Department of Computing, Goldsmiths, University of London
London, United Kingdom
{h.alhakbani, m.majid}@gold.ac.uk

Abstract. Over the years, machine learning has been facing the issue of imbalance dataset. It occurs when the number of instances in one class significantly outnumbers the instances in the other class. This study investigates a new approach for balancing the dataset using a swarm intelligence technique, Stochastic Diffusion Search (SDS), to undersample the majority class on a direct marketing dataset. The outcome of the novel application of this swarm intelligence algorithm demonstrates promising results which encourage the possibility of undersampling a majority class by removing redundant data whist protecting the useful data in the dataset. This paper details the behaviour of the proposed algorithm in dealing with this problem and investigates the results which are contrasted against other techniques.

Keywords: Swarm intelligence, support vector machine, class imbalance, stochastic diffusion search

One of the major issues in machine learning is the presence of imbalanced datasets. It occurs in most real world applications, including customers related dataset. When mining a customer data, potential buyers and churners are usually classified as minority class and data mining algorithms tend to be influenced by the majority class and misclassifying the minority class. In order for the classifiers to be accurately trained and not be influenced by the majority class, the datasets need to be balanced. In this work, a swarm intelligence algorithm (stochastic diffusion search or SDS) is used in order to undersample the majority class. The outcome is then analysed and the results are compared against other techniques. The research questions raised in this work are the following:

1. How does SDS help dealing with imbalanced datasets?
2. What are the other comparable techniques applicable to the problem of imbalanced datasets?
3. Does the application of SDS maintains the spread of the original data?
4. How does SDS deal with the computational cost of undersampling?

* Corresponding author

1 Related Work

In class imbalance [10], the difference in the numbers of the instances causes a skewed data distribution which affects the classifiers' capabilities in modelling some cases leading to low accuracy and bad classification model. Other factors affecting the classifier performance on imbalanced dataset are the sample size and difficulty of separating the minority class instances from the majority class instances [26]. Therefore, class imbalance issue has received attention in the literature [11, 14] and various types of solutions were proposed which include data level and algorithmic level solutions. This paper focuses on data level solutions where several solutions are proposed to overcome the class imbalance problem [9, 16, 20, 21, 8]. These include: random oversampling, random undersampling and a combination of both to balance the data [9, 20, 18, 4]. However, it has been argued that these techniques can result in removing useful information in the case of random undersampling, and over-fit in the case of random oversampling [9]. In spite of the abovementioned disadvantage, re-sampling are still popular to deal with class imbalance issue. Moreover, various researches have suggested better ways to both oversample and undersample such as Synthetic Minority Oversampling Technique (SMOTE) [8].

2 Stochastic Diffusion Search

Stochastic diffusion search SDS was first described by Bishop [6] as a population based matching algorithm that uses direct communication patterns such as co-operative transport found among social insects to perform evaluation of search hypothesis. Unlike other nature inspired search methods, SDS has a strong mathematical framework which describes the behaviour of the algorithm by investigating convergence to global continuum, linear time complexity and resource allocation [6, 25]. The SDS algorithm is applicable to changing the objective function, thus enabling a more robust response in dynamically changing environments. This feature of SDS makes the algorithm more attractive for various applications including but not limited to: eye tracking in facial images by using a combination of a Stochastic Search network and an n-tuple network [7], site decision for transmission equipment for wireless networks [17]; mouth locating in human faces images [15]; and more recently global optimisation and medical imaging applications [1]. A recent and comprehensive review of SDS details the key phases (test and diffusion phases) and use of this algorithm in the last two decades [2]. In the test phase each agent is labelled as active or inactive and in the diffusion phases agents communicate with each other (more details are provided in Section 3.1).

3 Experiments and Results

In this study, a set of experiments at the data level have been conducted to compare a few undersampling approaches which are applied to the direct marketing

campaigns of Portuguese bank dataset which can be accessed at the UCI Machine Learning Repository[1]. From the metadata, it has been found that there is a class imbalance which is caused by the dramatic difference between the number of subscribers that is equal to 451 and that of the non-subscribers which is equal to 3668. The proposed model uses support vector machine (SVM), which is one of the common and widely used classification algorithm. The model makes prediction using the radial kernel with Gamma set to 1.00 and the C set to 0.00. To prepare the dataset for SVM, the following pre-processing steps have been taken: all nominal values are converted to numerical; and all values are normalised to avoid the value scale difference among all attributes. As mentioned before, in order to oversample the minority class to 2000, SMOTE algorithm will be used. The aim is to oversample the minority class in order to reach a comparable size with the undersampled majority class.

3.1 Balancing the dataset

There are several methods to deal with the class imbalance problem in the dataset; the proposed model will investigate balancing the dataset by applying two different approaches: undersampling the majority class, and oversampling the minority class, which will be conducted using SMOTE.

The undersamping process is performed by SDS whose performance is then contrasted against random undersampling as well as undersampling with Euclidean distance, which will be described later in the paper.

Applying SDS: The initial work is to use SDS to undersample the majority class from 3668 to 2000 non-subscribers. In this experiment, the empirical value of 100 agents are used. Initially the model is selected from the search space (the entire non-subscribers) and the agents are set to find the closest match from the remaining items of the search space. Once a match or the most similar item is found, it is removed from the majority class with the aim of removing redundant data. Given this process aims at reducing the size of the search space without removing useful data, removing the closest item to a randomly selected model discourages the deletion of useful data. This hypothesis is later validated (in section 5.3) when the spread and the central tendency of the data are investigated before and after the undersampling process [22].

Following the initialisation phase where each agent is allocated to a hypothesis from the search space (a random non-subscriber), in the *test phase*, a randomly selected micro-feature (attribute) from the hypothesis is compared against the corresponding micro-feature of the model; if the randomly selected micro-feature of the hypothesis is within a specific threshold (which will be discussed later) from the model's micro-feature, the agent is set to active, otherwise inactive. This process is repeated for all the agents.

In the next phase, the *diffusion phase*, a passive recruitment mode is applied where each inactive agent chooses another agent and adopts the same hypothesis if the randomly selected agent is active. If the randomly selected agent is inactive,

[1] Link to the dataset: http://mlr.cs.umass.edu/ml/datasets/Bank+Marketing

the selecting agent picks a random hypothesis (i.e. a random non-subscriber from the search space). This process is repeated for all the inactive agents.

The cycle of test-diffusion is repeated 10 times, which is an empirically chosen value, at the end of which a non-subscriber with the maximum number of active agent is removed from the search space and the model is moved to another list (e.g. model list). This guarantees that while the most similar item is removed from the search space, the model, which represents the deleted item is kept and used later during the classification process. This process is repeated until the dataset is undersampled.

In the experiments reported in the paper, three different thresholds, including 1.00, 0.50 and 0.00 have been used and thus three different datasets of non-subscribers have been generated all sized 2000.

As the input dataset is normalised, SDS algorithm with threshold 1.00 *randomly* undersamples the data; threshold 0.00 looks for exact micro-feature match from the model; and threshold 0.50 is a state between random and exact-match undersampling.

Applying Euclidean Distance: Euclidean Distance (ED) is a metric used to measure distances between n points in the space. Over the past years, this measure has been widely used for database dimensionality reductions [19, 5]. Although a comprehensive metric, acknowledging the high computational expense of applying Euclidean distance to undersampling problem, it will be used in this work as the mean to contrast with the proposed cheaper computationally expensive swarm intelligence technique. ED will be used to undersample the majority class; in each iteration, a model is picked randomly, then the Euclidean distance of the model with each element in the search space is calculated; once all the distances are calculated, the closest element to the model is removed. This process process is repeated until the size of the search space is reduced to the number required (i.e. 2000 entries).

3.2 Results

In this work, various performance measures are used. Predictive accuracy is a widely used evaluation metric, however in the case of imbalanced data it is not in itself, a comprehensive evaluation tool as it does not show how the model correctly classifies the minority class, which is the class of interest [9]. To complement predictive accuracy rate, other performance metrics are also used: sensitivity, specificity, Area Under the Curve (AUC), f-measure and precision. The experimental results show that the new approach (i.e. a combination of SDS at threshold 0.00 to undersample the majority class, and SMOTE to oversample the minority class) achieves the best performance in terms of accuracy, specificity, F-measure and precision, as shown in Table 1. The proposed model achieves higher accuracy because of the higher specificity. On the other hand, obtaining higher F-measure is attributable to the higher precision rate as opposed to the euclidean distance undersampling. However when using euclidean distance for undersampling, the results exhibit higher sensitivity and AUC which can be justified given the much higher computational expense; this claim will be explored

Table 1: Performance Measurements comparison

Threshold	0.00	0.50	1.00	Euclidean Distance
Accuracy	**90.46**%	88.56%	88.56%	89.47%
Sensitivity	95.46%	96.06%	96.06%	**96.76**%
Specificity	**85.45**%	81.04%	81.04%	82.15%
AUC	0.959	0.96	0.96	**0.965**
F-measure	**90.93**%	89.41%	89.41%	90.91%
Precision	**86.82%**	83.67%	83.67%	84.48%

further in the next section along with a more in-depth discussion about SDS and the impact of the varying thresholds on the results. For all the experiments, 10-fold cross-validation is applied. Moreover, The results reported in this paper has shown that the proposed method can offer a promising results when compared against previous work on the same dataset (Portuguese bank dataset), as shown in table 2.

4 Discussion

In this section, the research questions raised in the introduction are discussed in the context of the experiments conducted and the results reported in the Section 3.

4.1 Applying SDS to imbalance data

SDS presents itself as an effective tool to persisting problems encountered in search and optimisation. This is due to SDS's strong partial function evaluation feature which assists the agents to explore the existing large search space and gather a global knowledge without having to evaluate all the existing dimensions; the in-depth analysis of the dimensions occurs only once a viable solution is found, at which stage, agents explore further dimensions of the plausible solutions.

In other words, the partial evaluation enables an agent to form a quick "opinion" about the quality of the investigated solution without exhaustive testing which potentially leads to increased computational complexity. However, using

Table 2: Results for Previous models on the direct marketing dataset

Models	AUC	Accuracy	Sensitivity	Specificity
Moro et al. [24]	0.938	NA	NA	NA
Moro et al. [23]	0.8	NA	NA	NA
G. Feng et al. [13]	NA	83%	NA	NA
Elsalamony [12]	NA	90.09%	59.06%	**93.23**%
Bahnsen et al. [3]	NA	88.28%	NA	NA
Proposed Model	**0.959**	**90.46%**	**95.46%**	85.45%

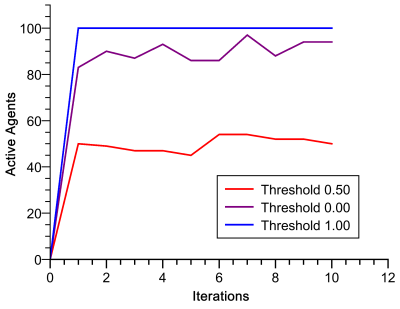

Fig. 1: SDS agent's activity

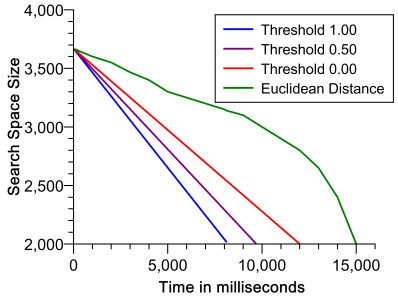

Fig. 2: SDS & EUC time comparison

euclidean distance, the search for a close match to the model is more computationally expensive because of the complete (vs. partial) function evaluation that accompanies each evaluation. Further discussion is presented about the computational cost of the presented technique in section 4.3.

In the experiments report earlier, three SDS threshold values are used. The activity of the agents in each of the three presented threshold are illustrated in Fig. 1. As expected, when the threshold is set to 1.00, all agents become active at the end of the first iteration, thus stopping communicating with other agents in the diffusion phase; when the threshold is set to 0.00, the algorithm only "settles" on an exact match, thus as shown in the figure, while around half of the agents are active, the other half searches for the closer match. The middle ground status of SDS when the threshold is set to 0.50, is also illustrated in the figure.

4.2 SDS and analysing the spread of data

Several metrics have already being used in the paper to evaluate the quality of the undersampled data during the classification phase. However another important metric to verify that useful data have not been removed during undersampling, is the spread of data and the central tendency. Using these measures, the mean and the standard deviation of all data points are calculated. The difference of each undersampled dataset from the original dataset (before undersampling) is 0.01095 ± 0.00925 and 0.00945 ± 0.00775 for SDS with threshold 0.0 and when using Euclidean distance respectively. The results highlight the lack of any significant change in the spread of data with threshold set to 0.00 and when euclidean distance is used. This shows not only the success of the algorithm in keeping the useful data, but also the presence of redundant data in the dataset.

4.3 SDS and computational complexity

As stated before, due to the partial function evaluation feature of SDS, the computational cost of running SDS on a huge dataset is only dependant on the number of agents and the number of iterations. In the presented work, where the initial size of the majority class is 3668, having 100 agents performing 10 iterations, means that the agents population partially evaluate more than quarter of

the search space (i.e. 100 agents $\times$ 10 iterations = 1000 micro-features evaluated). Overtime, with the shrinkage of the search space – thanks to the removal of the redundant data – the algorithm's coverage increases to half. One possible approach, which is the subject of an ongoing research, is to reduce the computational expense further by keeping the coverage of the swarm at a constant rate throughout the undersampling process. Fig. 2 shows the time taken for SDS in all three thresholds to undersample the data as well as when euclidean distance is used. As can be seen in the figure, the undersampling process with SDS demonstrate a linear time complexity throughout the undersampling process (where the search space is shrinking but the number of agents and the iterations allowed are constant), thus exhibiting the time complexity of $O(n)$.

5 Conclusion and Future Work

Various researches proposed advanced undersampling strategies to reduce the majority class samples without removing useful information. This work proposes a swarm intelligence based undersampling approach that reduces the sizes of the majority class in a reliable yet cheap computational way, using the agents and partial evaluation of the majority instance, in which the individuals of the swarm move through the solution space in search of solution that is close to the model. In the proposed method, the capability of SDS to perform majority class undersampling has been investigated on the real-world Portuguese bank dataset. The obtained results imply that SDS can be used as a good undersampling tool for class imbalance. Future work include the investigation of the SDS on other imbalanced dataset, as well as comparison with other swarm intelligence techniques that have been applied to overcome the class imbalance issue. Another topic of ongoing research is the relationship between (and the impact of) the population size of the SDS and the coverage percentage of the dynamically shrinking search space of the dataset being undersampled.

References

1. al-Rifaie, M.M., Aber, A., Sayers, R., Choke, E., Bown, M.: Deploying swarm intelligence in medical imaging. In: Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on. pp. 14–21. IEEE (2014)
2. al-Rifaie, M.M., Bishop, J.M.: Stochastic diffusion search review. Journal of Behavioral Robotics 3, 155–173 (2013)
3. Bahnsen, A.C., Aouada, D., Ottersten, B.: Ensemble of example-dependent cost-sensitive decision trees. arXiv preprint arXiv:1505.04637 (2015)
4. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM Sigkdd Explorations Newsletter 6(1), 20–29 (2004)
5. Beckmann, M., Ebecken, N.F., de Lima, B.S.P.: A knn undersampling approach for data balancing. Journal of Intelligent Learning Systems and Applications 7(04), 104 (2015)

6. Bishop, J.: Stochastic searching networks. In: Proc. 1st IEE Conf. on Artifical neural networks. pp. 329–331 (1989)
7. Bishop, J., Torr, P.: The stochastic search network. In: Neural networks for vision, speech and natural language, pp. 370–387. Springer (1992)
8. Chawla, N.V.: Data mining for imbalanced datasets: An overview. In: Data mining and knowledge discovery handbook, pp. 853–867. Springer (2005)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research pp. 321–357 (2002)
10. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. ACM Sigkdd Explorations Newsletter 6(1), 1–6 (2004)
11. Drown, D.J., Khoshgoftaar, T.M., Narayanan, R.: Using evolutionary sampling to mine imbalanced data. In: Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on. pp. 363–368. IEEE (2007)
12. Elsalamony, H.A.: Bank direct marketing analysis of data mining techniques. International Journal of Computer Applications 85(7) (2014)
13. Feng, G., Zhang, J.D., Liao, S.S.: A novel method for combining bayesian networks, theoretical analysis, and its applications. Pattern Recognition 47(5), 2057–2069 (2014)
14. García, V., Sánchez, J.S., Mollineda, R.A.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems 25(1), 13–21 (2012)
15. Grech-Cini, H., McKee, G.T.: Locating the mouth region in images of human faces. In: Optical Tools for Manufacturing and Advanced Automation. pp. 458–465. International Society for Optics and Photonics (1993)
16. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in intelligent computing, pp. 878–887. Springer (2005)
17. Hurley, S., Whitaker, R.M.: An agent based approach to site selection for wireless networks. In: Proceedings of the 2002 ACM symposium on Applied computing. pp. 574–577. ACM (2002)
18. Japkowicz, N., et al.: Learning from imbalanced data sets: a comparison of various strategies. In: AAAI workshop on learning from imbalanced data sets. vol. 68, pp. 10–15. Menlo Park, CA (2000)
19. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Knowledge and information Systems 3(3), 263–286 (2001)
20. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: ICML. vol. 97, pp. 179–186. Nashville, USA (1997)
21. Ling, C.X., Li, C.: Data mining for direct marketing: Problems and solutions. In: KDD. vol. 98, pp. 73–79 (1998)
22. McCluskey, A., Lalkhen, A.G.: Statistics ii: Central tendency and spread of data. Continuing Education in Anaesthesia, Critical Care & Pain 7(4), 127–130 (2007)
23. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decision Support Systems 62, 22–31 (2014)
24. Moro, S., Laureano, R., Cortez, P.: Using data mining for bank direct marketing: An application of the crisp-dm methodology (2011)
25. Nasuto, S.: Resource allocation analysis of the stochastic diffusion search. Ph.D. thesis, University of Reading (1999)
26. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23(04), 687–719 (2009)