Improving SMOTE with Fuzzy Rough Prototype Selection to detect Noise in Imbalanced Classification data

Nele Verbiest[1], Enislay Ramentol[2], Chris Cornelis[1,3], and Francisco Herrera[3]

[1] Dept. of Applied Mathematics and Computer Science, Ghent University, Belgium
Nele.Verbiest@UGent.be
[2] Dept. of Computer Science, University of Camagüey, Cuba
enislayr@yahoo.es
[3] Dept. of Computer Science and AI, University of Granada, Spain
chriscornelis@ugr.es
herrera@decsai.ugr.es

Abstract. In this paper, we present a prototype selection technique for imbalanced data, Fuzzy Rough Imbalanced Prototype Selection (FRIPS), to improve the quality of the artificial instances generated by the Synthetic Minority Over-sampling TEchnique (SMOTE). Using fuzzy rough set theory, the noise level of each instance is measured, and instances for which the noise level exceeds a certain threshold level are deleted. The threshold is determined using a wrapper approach that evaluates the training Area Under the Curve of candidate subsets. This proposal aims to clean noisy data before applying SMOTE, such that SMOTE can generate high quality artificial data.
Experiments on artificial data show that FRIPS in combination with SMOTE outperforms state-of-the-art methods, and that it particularly performs well in the presence of noise.

Keywords: SMOTE, Imbalanced Classification, AUC, Fuzzy Rough Set Theory

1 Introduction

Imbalanced classification has become an important field in data mining. In contrast to traditional classification, it deals with datasets where one or more classes are under-represented. In this paper we consider the two-class case where one class (the majority or negative class) is over-represented and the other class (the minority or positive class) is under-represented. The Imbalance Ratio (IR, the size of the majority class divided by the size of the minority class) characterizes the imbalance of the datasets: a dataset with IR 1 is perfectly balanced, while datasets with a higher IR are more imbalanced.
Standard data mining techniques might not always work well for the imbalanced problem, as their results are often biased towards the majority class. One

cause for this is that data mining techniques are often based on global quantities like classification accuracy. Instead of classification accuracy, one may use the Receiver Operating Characteristic (ROC) curve. It plots the ratio of correctly classified minority instances against the ratio of correctly classified majority instances. As a result, the Area Under the ROC Curve (AUC, [2]) can be used to evaluate data mining techniques for imbalanced data. It reflects the trade-off between correctly classified minority and majority instances.

Many techniques have been developed for imbalanced data, both on the classifier level and on the data level. In this work we focus on the data level, i.e., on preprocessing techniques that transform datasets such that they are better suited for the imbalanced classification task. More specifically, we focus on the Synthetic Minority Over-sampling TEchnique (SMOTE, [4]) that forms new minority instances by interpolation to balance the dataset.

As SMOTE is sometimes too forceful in adding new minority instances, some improvements on the technique have been studied. E.g., Borderline-SMOTE [11] only re-samples border instances, while SMOTE with Tomek links and SMOTE-ENN [1] apply data cleaning after re-sampling the minority instances.

In this paper, we present another improvement of SMOTE that cleans the data before applying SMOTE, that is, we try to remove noisy instances from the data, such that the quality of the instances introduced by SMOTE is better. As there is no reason to assume that only the majority instances can be noisy, we both remove minority and majority instances.

We proceed as follows: for each instance, we calculate its noise level using a measure based on fuzzy rough set theory [7]. Next, we remove all instances that have a noise level higher than a certain threshold, which is determined by a wrapper procedure that evaluates different thresholds based on the training AUC of the resulting subsets of instances. After this data preprocessing, we finally apply SMOTE. We call this technique Fuzzy Rough Imbalanced Prototype Selection (FRIPS) and call it FRIPS-SMOTE when it is applied in combination with SMOTE.

Note that we use the term *prototype* selection instead of *instance* selection. The reason for this is that our technique is specifically developed to improve the K Nearest Neighbor (KNN, [5]) classifier. In the KNN context, instance selection is often called prototype selection [8].

We use KNN because it is a simple classification method that does not impose assumptions on the data. Due to its local nature it has low bias, more specifically, the error rate of 1NN asymptotically never exceeds twice the optimal Bayes error rate We use the 1NN classifier as this classifier is most susceptible to noisy data.

The remainder of this paper is structured as follows: In Section 2.1 we introduce a noise measure based on fuzzy rough set theory, that is used in the FRIPS algorithm introduced in Section 2.2. In Section 3.1 we describe the set-up of the experimental evaluation. In Section 3.2 we present the results of the experimental evaluation, which show the good performance of FRIPS. We conclude and suggest future research directions in Section 4.

2 Fuzzy Rough Imbalanced Prototype Selection

2.1 A Noise Measure based on Fuzzy Rough Set Theory

We consider a decision system $(X, \mathcal{A} \cup \{d\})$ that consists of a set of instances $X = \{x_1, \ldots, x_n)$, a set of continuous attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$ and a fixed decision attribute $d \notin \mathcal{A}$. The value of an attribute $a \in \mathcal{A}$ for an instance $x \in X$ is denoted by $a(x)$, and we assume that these values are normalized, that is, $a(x) \in [0, 1]$ for all $x \in X$ and $a \in \mathcal{A}$. The class of each instance is denoted by $d(x)$ and can take two values: 0 or 1.
In [16], the following measure was introduced, based on fuzzy rough set theory, to express how noisy an instance $x \in X$ is [7]:

$$\forall x \in X : \alpha(x) = \underbrace{OWA_W \frac{1}{\sum_{i=1}^{m} \delta_{a_i}(x, y)}}_{y \in \{z \in X | d(x) \neq d(z)\}}. \tag{1}$$

In this formula, for each attribute $a \in \mathcal{A}$, δ_a is a distance measure defined as follows:

$$\forall x, y \in X : \delta_a(x, y) = (a(x) - a(y))^2. \tag{2}$$

As we assume that all attributes are normalized, this distance returns a value between 0 and 1.
The OWA_W[18] operator is an aggregation operator that, given a series of values $a_1, \ldots, a_p \in \mathbb{R}$ and a weight vector $W = \langle w_1, \ldots, w_p \rangle$ that fulfills $\forall i \in 1, \ldots, p : w_i \in [0, 1]$ and $\sum_{i=1}^{p} w_i = 1$, is given by:

$$OWA_W(a_1, \ldots, a_p) = \sum_{i=1}^{p} w_i b_i, \tag{3}$$

where $b_i = a_j$ if a_j is the ith largest value in $a_1, \ldots, a_p$. That is, the values are ordered and then a weighted average is applied to these values. In our case, the weights are defined by:

$$\forall i \in 1, \ldots, p : w_i = \frac{2(p - i + 1)}{p(p + 1)}. \tag{4}$$

As these weights are decreasing, OWA_W is a softening of the maximum operator, which can be represented by the weight vector $(1, 0, \ldots, 0)$.
Note that the noise value $\alpha(x)$ is proportional to the distance to instances from other classes. When many instances from other classes are close to x, the noise value will be high.

2.2 Fuzzy Rough Imbalanced Prototype Selection

The noise measure described in the previous subsection can be used to apply prototype selection: instances with a low noise value should be retained, while

instances with high noise values should be removed. The difficulty is now to find a good threshold for the noise values.

The FRIPS algorithm proceeds as follows: the noise values of all instances are considered and any of these values is used as threshold. Each threshold corresponds to a subset of instances, namely those instances that have a noise value not higher than the threshold. These subsets are evaluated by measuring their training AUC. The threshold corresponding with the subset of instances that has the best training AUC is finally selected. In case there is more than one optimal threshold, the median of all thresholds is chosen.

In Algorithm 1, the procedure that calculates the training AUC is given. As we use the 1NN algorithm [5] in the experiments as final classifier, we also use this classification rule in the FRIPS procedure. The confusion matrix C is initialized in Line 2. Then we classify all instances in X using the leave-one-out procedure: to classify a training instance x w.r.t. a subset S of all training instances, we look up the nearest neighbor of x in the entire set S if x is not contained in S, and in $S \setminus \{x\}$ otherwise. For each classified instance we update C and at the end we calculate the AUC based on C.

The final FRIPS procedure is described in Algorithm 2. In Line 2 and 3, the candidate noise thresholds are calculated. In Line 4 to 6, the training AUC of the complete training set is calculated. In order to do that, the nearest neighbors of all instances need to be calculated. In the loop going from Line 8 to 19, the candidate thresholds are evaluated. As we evaluate them in decreasing order, the nearest neighbors do not need to be re-calculated for all instances in each iteration: only the instances that have neighbors that are removed in Line 9 need to be re-calculated. As a result, we can keep the running-time of the FRIPS algorithm under control. In Line 20, the final noise threshold is selected and the instances that have a noise value lower than or equal to this threshold are returned in Line 22.

Algorithm 1 trainAUC, procedure to measure the AUC of a subset of instances using a leave-one-out approach.

1: **input**: Reduced decision system $(S, \mathcal{A} \cup \{d\})$ $(S \subseteq X)$.
2: Initialize confusion matrix $C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
3: **for** $x \in X$ **do**
4: **if** $x \in S$ **then**
5: Find the nearest neighbor nn of x in $S \setminus \{x\}$
6: $C(d(x), d(nn)) \leftarrow C(d(x), d(nn)) + 1$
7: **else**
8: Find the nearest neighbor nn of x in S
9: $C(d(x), d(nn)) \leftarrow C(d(x), d(nn)) + 1$
10: **end if**
11: **end for**
12: **Output** AUC based on C.

Algorithm 2 FRIPS

1: **input**: Decision system $(X, \mathcal{A} \cup \{d\})$
2: Calculate $\alpha(x_1), \ldots, \alpha(x_n)$
3: Remove duplicates and order the α values from step 2: $\alpha_1 > \alpha_2 > \ldots > \alpha_p$
4: opt.alphas $\leftarrow \{\infty\}$
5: Calculate nearest neighbors of all instances
6: auc.opt $\leftarrow trainAUC(X, \mathcal{A} \cup \{d\})$
7: auc.current $\leftarrow$ auc.opt
8: **for** $\alpha = \alpha_2, \ldots, \alpha_p$ **do**
9: Remove instances x for which $\alpha(x) > \alpha$, the resulting set of instances is S
10: **if** Number of remaining instances > 1 **then**
11: Recalculate nearest neighbors of instances for which current nearest neighbor was removed in step 9
12: auc.current $\leftarrow trainAUC(S, \mathcal{A} \cup \{d\})$
13: **if** auc.current $>$ auc.opt **then**
14: opt.alphas $\leftarrow \{\alpha\}$
15: **else if** auc.current $=$ auc.opt **then**
16: opt.alphas $\leftarrow$ opt.alphas $\cup\{\alpha\}$
17: **end if**
18: **end if**
19: **end for**
20: best.alpha = median(opt.alphas)
21: Remove instances x for which $\alpha(x) > best.alpha$, the resulting set of instances is S
22: **Output** $(S, \mathcal{A} \cup \{d\})$

3 Experimental Study

In this section we evaluate the performance of our algorithm. In Section 3.1 we present the datasets used for the experimentation and list the algorithms to which we compare our algorithm. In Section 3.2 we present and discuss the obtained results.

3.1 Experimental Set-Up

We use the datasets that were constructed by Napierała et al. in [12]. All datasets are binary and are randomly and uniformly distributed in a two-dimensional feature space. The minority class takes three different shapes in the feature space: the *subclus* data has 3 rectangles of minority instances, in the *clover* data the minority instances form a flower with five elliptic petals, and the *paw* datasets have three elliptic subregions of minority instances, of which two subregions are close to each other. The datasets are constructed with 600 or 800 instances. In case of 600 instances, the IR is 5, in case of 800 instances, the IR is 7.
To test if FRIPS can handle noise, we use the same data, where the borders

of the subregions in the minority class were disturbed. The Disturbance Ratio (DR) is 0, 30, 50, 60 and 70 % , where the DR is the ratio of the width of the overlapping minority subregion compared to the total width of the subregion. As a result, there are 30 datasets: there are three shapes, with 600 or 800 instances and 5 DR levels.
We compare our algorithms to several state-of-the-art approaches. We consider the SMOTE algorithm itself and the following improvements of it:

- SMOTE with data cleaning using Tomek Links (SMOTE-TL [1])
- SMOTE with data cleaning using the Edited Nearest Neighbour technique (SMOTE-ENN [1])
- Borderline SMOTE, where only border instances are re-sampled (SMOTE-BL1[11])
- A variation on SMOTE-BL1 where the synthetic instances are closer to the minority class (SMOTE-BL2[11])
- SMOTE weighting the minority instances according to their safe-level (SMOTE-SL [3])
- SMOTE with data cleaning using Rough Set Theory (SMOTE-RSB,[13])
- SMOTE with data cleaning using Fuzzy Rough Set Theory (SMOTE-FRS, [14])

Furthermore, we also consider the SPIDER [15] algorithm, which removes majority instances that result in misclassifying instances from the minority class and over-samples minority instances that are surrounded by majority instances [15]. The last algorithm we use is SPIDER2: a two-phase version of the SPIDER algorithm presented in [12]. In the first phase noisy majority instances are removed or relabeled, in the second phase noisy minority examples are amplified.
We use a 5 fold cross validation strategy: each dataset is divided in 5 folds, and the instances of each fold (test data) are classified using the remaining folds as training data. The training data is preprocessed using the state-of-the-art techniques, FRIPS and FRIPS-SMOTE, and afterwards the test data is classified using the 1NN rule applied on the training data. We report the average AUC over all test folds. All procedures are implemented in the Keel[4] software platform.

3.2 Results

In Figure 1, the average AUC values over all 30 datasets are given for each method. It can be seen that all preprocessing techniques improve the KNN classification. The state-of-the-art techniques SMOTE-ENN and SMOTE-TL improve SMOTE quite well. On the other hand, SMOTE-RSB and SMOTE-FRS, both techniques that try to improve SMOTE by deleting instances from the dataset processed by SMOTE, do not improve SMOTE. FRIPS improves SMOTE but is not better than SMOTE-TL. On the other hand, if we use FRIPS to clean the data before applying SMOTE, we obtain very good results.

[4] www.keel.es

To see if FRIPS-SMOTE significantly outperforms the state-of-the-art results, we perform the statistical Wilcoxon test [17]. This is a non-parametric pairwise test that aims to detect signifcant differences between two sample means; that is, the behavior of the two implicated algorithms in the comparison. For each comparison we compute $R+$, the sum of ranks of the Wilcoxons test in favor of FRIPS-SMOTE, $R-$, the sum of ranks in favor of the other methods, and also the p-value obtained for the comparison. The observed values of the statistics are listed in Table 1. As the p-value is always lower than 0.05, we can conclude that FRIPS-SMOTE outperforms all state-of-the-art algorithms at the 5% significance level.

Besides, we perform a statistical analysis conducted by non-parametric multiple comparison procedures [6, 10, 9]. We use Friedman's procedure to compute the set of ranks that represent the effectiveness associated with each algorithm. In addition, we compute the adjusted p-value with Holm's test. The Friedmann rankings are given in Table 1, together with Holm's adjusted p-values. FRIPS-SMOTE obtains the highest ranking and outperforms all algorithms except SMOTE-TL at the 5% significance level.

Next, we analyze what the effect of the border disturbance is on the performance of FRIPS-SMOTE. Therefore, we compare it to the two best-performing state-of-the-art algorithms: SMOTE-ENN and SMOTE-TL. In Figure 2, the results are depicted for each dataset depending on the border disturbance ratio. From this, we see that FRIPS-SMOTE performs more or less equally well as the other algorithms if no border noise is added, but that it performs better if an intermediate amount of border noise is added. It must also be noted that all methods are highly susceptible to the border disturbance: there is a drop of about 10 % AUC.

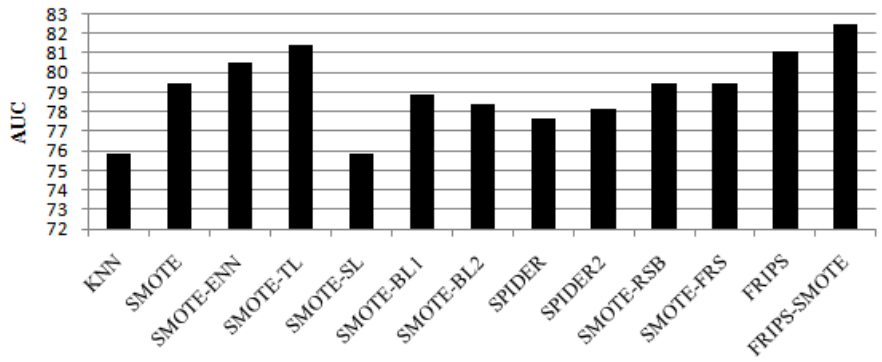

Fig. 1. Average AUC values over all datasets for each method.

Table 1. Observed values of the Wilcoxon test, Friedman test and Holm's post-hoc test, comparing FRIPS-SMOTE to state-of-the-art algorithms.

	Wilcoxon			Friedman	Holm
FRIPS-SMOTE vs.	R+	R-	p-value	Friedman Ranking	p-value
KNN	459.0	6.0	0.000003	10.9	0
SMOTE	400.0	35.0	0.000076	5.2167	0.012532
SMOTE-ENN	386.0	79.0	0.001537	4.6833	0.043861
SMOTE-TL	349.0	116.0	0.015566	3.4	0.361218
SMOTE-SL	459.0	6.0	0.000003	11.0333	0
SMOTE-BL1	432.0	33.0	0.000036	6.4833	0.000143
SMOTE-BL2	424.0	41.0	0.000078	6.8167	0.000032
SPIDER	452.0	13.0	0.000006	8.4833	0
SPIDER2	455.0	10.0	0.000004	7.5833	0.000001
SMOTE-RSB	417.0	48.0	0.000136	5.3167	0.011839
SMOTE-FRS	418.5	46.5	0.00012	5.5333	0.006762
FRIPS-SMOTE	-	-	-	2.55	-

4 Conclusion and Future Work

In this paper, we have presented a new improvement of the SMOTE oversampling technique, FRIPS-SMOTE. It cleans the data before applying SMOTE by measuring the noise of every instance using fuzzy rough set theory, and selecting a noise threshold using a wrapper approach that evaluates candidate thresholds w.r.t. the corresponding training AUC.
Experiments on artificial data show that FRIPS-SMOTE outperforms state-of-the-art methods, and that it particularly performs well if the borders of the minority classes are disturbed.
In the future we want to take this work a step further by applying further data cleaning techniques on the dataset preprocessed by FRIPS-SMOTE. Moreover, we want to experiment with other prototype selection techniques, and we want to study the impact of FRIPS-SMOTE on real datasets.

Acknowledgment

This work was partially supported by the P11-TIC-7765 project.

References

1. G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.
2. A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.

3. C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD09)*, volume 5476, pages 475–482, 2009.
4. N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
5. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
6. Joaqun Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
7. D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, 17:191–209, 1990.
8. S. García, J. Derrac, J.R. Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.
9. S. García, A. Fernández, J. Luengo, and F. Herrera. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, 13:959–977, 2009.
10. S. García, J. Alcalá Fernandez, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180:2044–2064, 2010.
11. H. Han, W. Wang, and B. Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *ICIC (1)*, pages 878–887, 2005.
12. K. Napierala, J. Stefanowski, and S. Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *7th International Conference on Rough Sets and Current Trends in Computing(RSCTC2010)*, pages 158–167, 2010.
13. E. Ramentol, Y. Caballero, R. Bello, and F. Herrera. Smote-rsb*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and Information Systems*, 2011.
14. E. Ramentol, N. Verbiest, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera. Smote-frst: A new resampling method using fuzzy rough set theory. In *10th International FLINS Conference on Uncertainty Modeling in Knowledge Engineering and Decision Making (FLINS 2012)*.
15. J. Stefanowski and S. Wilk. Selective pre-processing of imbalanced data for improving classification performance. In *10th International Conference in Data Warehousing and Knowledge Discovery(DaWaK2008)*, volume 5182 of *Lecture Notes on Computer Science*, pages 283–292. Springer, 2008.
16. N. Verbiest, C. Cornelis, and F. Herrera. Fuzzy rough prototype selection. Submitted.
17. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, (6):80–83, 1945.
18. R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190, 1988.

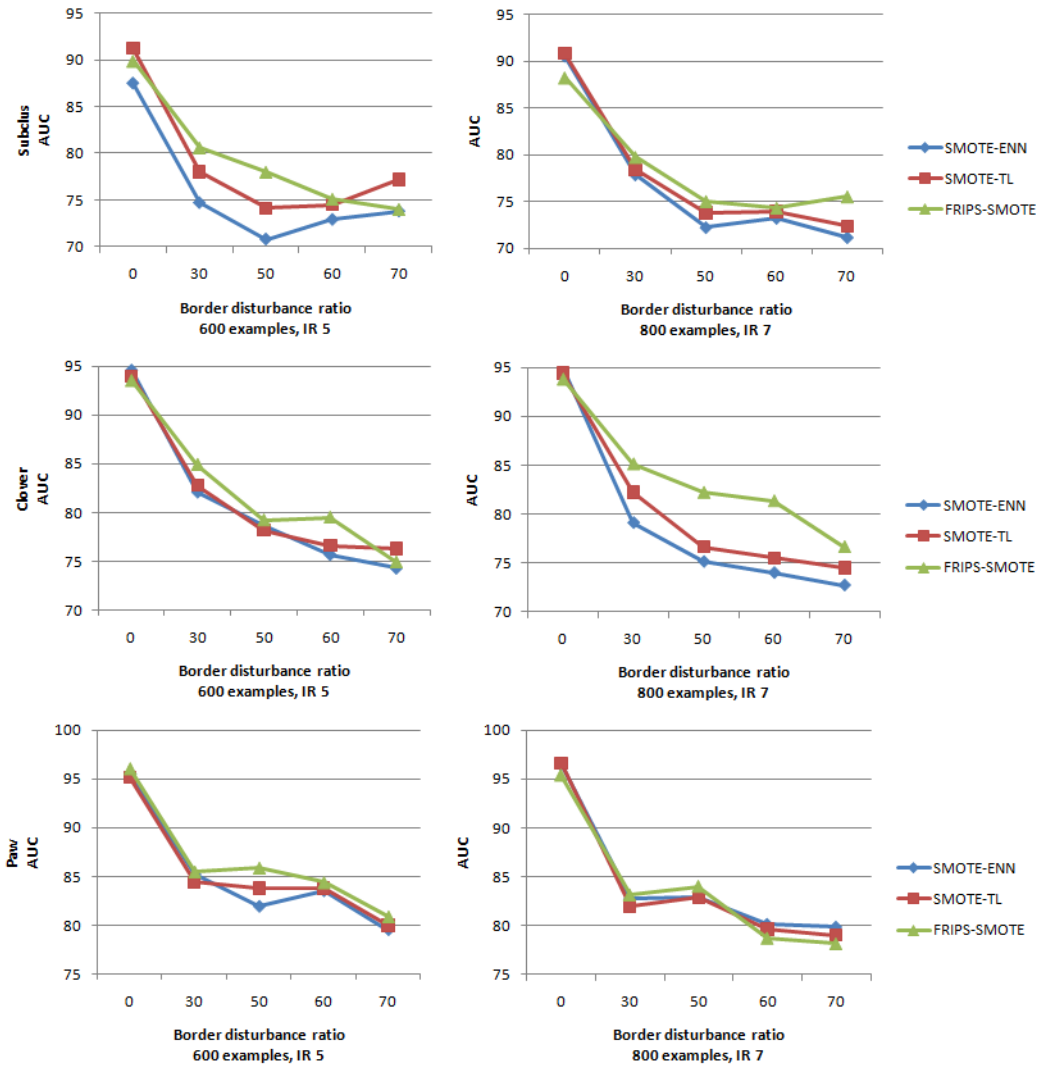

Fig. 2. Results for each dataset, comparing the results with different border disturbance ratios.