Evaluating the Impact of Data Quality on Sampling

Jason Van Hulse (jvanhulse@gmail.com)
Taghi M. Khoshgoftaar (taghi@cse.fau.edu)
Amri Napolitano (anapoli1@fau.edu)
Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University
777 Glades Rd., Boca Raton, FL 33431

***Abstract*—Three important data characteristics that can substantially impact a data mining project are class imbalance, poor data quality and the size of the training dataset. Data sampling is a commonly used method for improving learner performance when data is imbalanced. However, little effort has been put forth to investigate the performance of data sampling techniques when data is both noisy and imbalanced. In this work, we present a comprehensive empirical investigation of how data sampling techniques react to changes in four training dataset characteristics: dataset size, class distribution, noise level and noise distribution. We present the performance of four common data sampling techniques using 11 learning algorithms. The results, which are based on an extensive suite of experiments for which over 15 million models were trained and evaluated, show that data sampling can be very effective at dealing with the combined problems of noise and imbalance. In addition, the dataset characteristics which have the greatest impact on each of the data sampling techniques are identified.**

I. Introduction

Imbalanced training datasets are common among many application domains. A dataset is said to be imbalanced if examples of one class greatly outnumber examples of the other class(es). Such a situation can make constructing useful classification models difficult. Many traditional machine learning algorithms are designed to maximize overall accuracy without regard for the significance of the individual classes. When data is highly imbalanced, or skewed, such learners tend to favor the majority (or negative) class, misclassifying a disproportionately high number of minority (or positive) class examples.

Class imbalance alone may not hinder performance unless characteristics of the data (e.g., difficult to learn concepts) make it more challenging to differentiate between examples of the various classes. If the training dataset contains easy to learn concepts and high quality data (i.e., low noise levels), then a model may achieve nearly 100% accuracy, even when the data contains severe class imbalance. Such a dataset is very rare in practice. A common data quality factor that can make concepts more difficult to learn is data noise.

Noise can be categorized as either class noise or attribute noise. Class noise refers to mislabeled examples, i.e., examples that are labeled as one class but really belong to another class. Attribute noise refers to incorrect values in the independent (non-class) attributes. Noise can be the result of several factors including poor database management or data collection procedures. Both types of noise have been shown to negatively impact classification performance, but class noise generally has the greater impact on learning [1]. Our study also considers the noise distribution within the data, something few studies have investigated [2].

Data sampling techniques have been proposed to alleviate the problems associated with learning from imbalanced training data. These techniques artificially balance the class distribution by either augmenting the minority class (oversampling) or removing examples from the majority class (undersampling). This study considers four of the most common procedures: random undersampling (RUS), random oversampling (ROS), Wilson's Editing (WE) [3] and SMOTE [4].

We present a comprehensive empirical investigation of learning from noisy and imbalanced data, for which over 15 million models were trained and evaluated. The objective of this work is to understand which data sampling techniques perform best when data is noisy and imbalanced, and to identify which factors have the greatest impacts on each of these techniques. Models are trained using 11 different learners from various machine learning paradigms. The results show that each data sampling technique can improve performance, but the best sampling technique often depends on the size, class distribution, noise level and noise distribution of the data.

II. Related Work

While much research has been conducted examining the problems of class imbalance and noisy data in isolation, the combined effects of these factors have not received sufficient attention [5]. Class imbalance has been considered in many studies [6], [7]. A commonly-used method for improving the performance of classifiers when data is imbalanced is data sampling. Sampling techniques, which attempt to balance class distributions by adding examples to, or removing examples from, the training data, have received significant attention [8], [9], [10]. The majority of this research, however, is limited to the topic of class imbalance and does not address quality of data issues. In this work, we examine four data sampling techniques, finding that they each have their strengths and weaknesses depending on various characteristics of the training data. One related work [11] that does consider sampling in the context of noisy and imbalanced data differs from these experiments in a few important aspects. First, this current work utilizes different datasets and varies two additional experimental dimensions, the training dataset size and the class distribution. This study utilizes a different performance metric, the F-Measure, which as will be shown is critical since the conclusions derived based on this metric are quite different than that of the area under the ROC curve. Finally, this work focuses more on the impact of data quality and data availability on the individual sampling techniques across learning algorithms.

Numerous studies have been performed to analyze the impact of noise and how to cope with it. Three general methods have been proposed to alleviate the problem of noise in data. The first is to use a robust classification algorithm [12]. A classification algorithm is said to be robust in the presence of noise if it is able to maintain a high classification accuracy despite the low quality of data. Another common technique to overcome the problem of noisy data is to apply a filter designed to remove noisy instances from the dataset [13], [14], [15]. The third technique is to attempt to correct the noisy data [16]. Noise correction is often considered superior to filtering, especially in small datasets where the removal of records could result in the loss of important data which could be beneficial to the learning process.

IEEE IRI 2010, August 4-6, 2010, Las Vegas, Nevada, USA
978-1-4244-8099-9/10/$26.00 ©2010 IEEE

	LetterA	Nursery3	OptDigits8	Splice2
Size	20000	12960	5620	3190
#min	789	328	554	768
%min	3.95	2.53	9.86	24.08

TABLE I
CHARACTERISTICS OF INITIAL DATASETS

III. EXPERIMENTS

A. *Experimental Datasets*

The experiments in this work are based on four datasets, all acquired from the UCI Repository [17]. Each of the datasets originally contained multiple class values, but were transformed into binary class problems. The datasets selected for use in these experiments, as well as information about their sizes and class distributions, can be found in Table I. These datasets were utilized because they are relatively clean (achieved average AUC values near 1 using all 11 learners, and ten runs of 10-fold cross validations) and their sizes and class distribution facilitate our experiments as described in Sections III-B1 through III-B4.

B. *Experimental Design*

Using the four datasets described in Section III-A, we derive several subsets with different sizes, class distributions, noise levels and noise distributions for use in our experimentation.

1) Experimental Factor: Dataset Size: The first dataset characteristic (experimental factor) considered is the dataset size (denoted DS). Based on the original datasets, subsets of the data are selected to use as training datasets. Three sizes for these training subsets are used (DS={1500, 3000, 4500}), where the value of DS indicates the number of examples in the training subset. For example, a dataset with DS=1500 has 1500 examples. Due to constraints related to the original dataset sizes and class distributions, only LetterA and OptDigits8 are used to create datasets of all three sizes. In this work, when the impact of dataset size is considered, only datasets subsets from LetterA and OptDigits8 are used. When not considering dataset size, all four datasets are used with DS=1500.

2) Experimental Factor: Class Distribution: The second experimental factor, class distribution (CD), indicates the percentage of examples in the training subsets belonging to the minority class. The experiments in this work consider six levels of class distribution CD={1,2,4,6,8,10}, where the value of CD indicates the percentage of randomly selected examples in the training data that belong to the minority class. For example, a training subset with CD=4 contains 4% of its examples from the minority class, and 96% examples from the majority class (examples are selected at random to obtain the appropriate class distribution). Note that this is the ratio prior to noise injection (discussed in the following sections). Depending on the noise distribution, the final subset of training data may contain a different class distribution.

3) Experimental Factor: Noise Level: The third factor, noise level (NL), determines the quantity of noisy examples in the training data. The selected datasets are relatively clean, so NL is varied by artificially injecting noise into the dataset. This is accomplished by swapping the class value of some of the examples. The number of examples with their classes swapped is a function of NL and the number of minority examples in the training subset.

While many works involving noise injection often inject noise by simply selecting $x\%$ of the examples and corrupting their class, this technique is inapropriate when dealing with imbalanced datasets. For example, if a dataset contains only 1% of its examples belonging to the minority class, and as little as 10% of its examples are corrupted (injected with noise), the minority class will become overwhelmed by noisy examples from the majority class. Instead, we corrupt a percentage of the examples based on the number of minority examples in the dataset.

In our experiments, we use five levels of noise, NL={10,20,30,40,50}, where NL determines the number of examples, based on the size of the minority class, that will be injected with noise. The actual number of noisy examples will be:

$$2 \times \frac{NL}{100} \times P$$

where P is the number of positive (minority) examples in the dataset. For example, a dataset with DS=1500, CD=10 and NL=20, will have 150 minority examples (10% of 1500) and $2 \times 0.2 \times 150 = 60$ noisy examples. Note that this does not indicate which examples (minority or majority) will be corrupted. That is determined by the final experimental factor: noise distribution.

4) Experimental Factor: Noise Distribution: The final experimental factor, noise distribution ND, determines the type of noise that will be injected into the data. When dealing with binary class datasets (the only kind considered in this work) there are two possible noise types: P→N and N→P. Noise of type N→P occurs when an example that should be labeled "negative" is incorrectly labeled as "positive." Conversely, P→N noise occurs when an example that should be labeled "positive" is instead labeled "negative."

Five levels of noise distribution are used (ND={0,25,50,75,100}), where the value of ND indicates the percentage of noisy examples that are of type P→N. For example, if a training subset is to contain 60 noisy examples, and ND=25, then 25% (15) of those noisy examples will be minority ("positive") class examples that have their labels changed to "negative" (P→N), while the remaining 75% (45) of the noisy examples will be of type N→P. Due to the definition of ND, the combination of ND=100 and NL=50 can not be used, since the resulting dataset would have zero minority examples.

When analyzing the impact of NL on learner performance, a noise distribution of ND=50 will often be used. This distribution is known as PRC (Proportional Random Corruption [18]). This noise distribution is important since the class distribution after noise injection is the same as before noise is injected.

C. *Performance Measurement*

Two different performance metrics are utilized in our experiments: the area under the ROC curve and the F-measure. Receiver Operating Characteristic curves, or ROC curves, graph true positive rates on the y-axis versus the false positive rates on the x-axis. The ROC curve illustrates the performance of a classifier across the complete range of possible decision thresholds. The threshold independent nature of ROC curves makes them very well suited for describing the classification performance of models built on imbalanced data. The area under the ROC curve (AUC) provides a single numerical value representing the strength of a model as measured in ROC space.

The other performance metric used in this work is the *F-Measure*. This metric is derived from *recall* (or true positive rate) and *precision*. The use of precision (rather than false positive rate) provides a different view of classification performance than ROC curves. The formula for F-Measure, which is given in Equation 1, uses a tunable parameter, β, to indicate the relative importance of recall and precision. That is, one can modify β to place more emphasis on either recall or precision. Typically, $\beta = 1$ is used (as is the case in our study).

$$\text{F-Measure} = \frac{(1+\beta^2) \times recall \times precision}{\beta^2 \times recall + precision} \quad (1)$$

D. Learners

All results in this study are presented as the average of 11 commonly used learners from various modeling paradigms including neural networks, decision trees, regression modeling, Bayesian learning and case-based reasoning. All learners are implemented in Weka [19]. By averaging the performance of a wide variety of learners, we can focus our attention on the performance of the sampling techniques in general, without considering the interaction between learning and sampling techniques. That is, our goal is to identify the general performance of each sampling technique, regardless of learner.

The learners consist of two nearest neighbor techniques (2NN and 5NN), two C4.5-based techniques (C4.5D and C4.5N), Logistic Regression (LR), two types of artificial neural networks (MLP and RBFNet), Naive Bayes (NB), Random Forest (RF), RIPPER and Support Vector Machines (SVM). The parameters for the C4.5D, LR, NB, RF and RIPPER classifiers were kept at the default values in Weka. C4.5N disables pruning and enables Laplace smoothing. 2NN and 5NN use two and five nearest neighbors, respectively, and the distanceWeighting parameter was set to Weight by 1/distance. For the MLP learner, the hiddenLayers parameter was changed to 3 to define a network with one hidden layer containing three nodes, and the validationSetSize parameter was changed to 10 to cause the classifier to leave 10% of the training data aside to be used as a validation set to determine when to stop the iterative training process. For the RBFNet learner, the 'numClusters' parameter was changed to 10. For SVM, the complexity constant 'c' was changed from 1.0 to 5.0 and the 'buildLogisticModels' parameter was enabled. Default parameters in Weka were modified only when such changes generally resulted in superior performance across a wide range of experimental settings. No attempt was made to optimize these parameters, but only to set reasonable values for the purposes of our experiments. Future work can consider different learners or parameter values.

E. Sampling Techniques

The learners listed above are performed both with and without the use of data sampling. In this work, four well-known data sampling techniques are used. The oversampling techniques used are random oversampling (ROS) and SMOTE (SM). ROS simply duplicates minority class examples randomly to achieve a more balanced class distribution, while SM creates new minority class examples. The undersampling techniques used are random undersampling (RUS) and Wilson's editing (WE). RUS randomly removes examples from the majority class to achieve a more balanced class distribution, while WE tries to remove only noisy examples. RUS, ROS and SM each require an input parameter indicating the desired minority class percentage in the post-sampling dataset. We use values of 35%, 50% and 65% and only include the best parameter selection in our results. Selecting an optimal post-sampling class distribution remains an open research issue and is outside the scope of this work. Also, WE is performed using the Euclidean and Weighted distance measures [20], and we select only the best of the two for analysis.

F. Experimental Procedure

All experiments are performed using 10-fold cross validation. The data is split into ten partitions, nine of which are used to train models, while the remaining partition is used as test data. All experimental parameters are applied only to the training portion of the data. The entire test dataset (1/10 of the original dataset, as described in Table I) is used to test the models, and is never injected with noise or modified in any way. Only the training data is sampled to achieve the desired dataset size and class distribution, then injected with noise at the appropriate level and distribution.

Four data sampling techniques are applied to each training dataset, using different parameters. RUS, ROS and SMOTE are each performed at three different levels, 35, 50 and 65, while WE is performed using two different versions. Models are also trained without sampling (denoted None) as a baseline for comparison. In total, $3 \times 3 + 1 \times 2 + 1 = 12$ sampling technique/parameter combinations are performed in this work.

The dataset creation process is repeated ten times so that each fold acts as test data once, and ten runs of the entire process (i.e., ten independent runs of ten-fold cross validation) are performed to eliminate any biasing that may occur during the random partitioning, sampling and noise injection processes. Therefore each experiment (combination of dataset, learner, dataset size, class distribution, noise level and noise distribution) is repeated 100 times. For all 4 datasets, 6 class distributions and 24 combinations of noise level and noise distribution are used. For 2 of the datasets (LetterA and OptDigits8), 2 additional dataset sizes are used, with the same 6 class distributions and 24 noise combinations. Using 11 learners and 12 data sampling/parameter combinations, the total number of experiments performed was $11 \times 12 \times ((4 \times 6 \times 24) + (2 \times 2 \times 6 \times 24)) = 152,064$, each of which is repeated 100 times (ten runs of 10-fold cross validation), resulting in a total of over 15 million models built and evaluated for this work.

IV. Empirical Results

This section presents the results of the experiments described in Section III. The objective of this work is to identify how variations in class distribution, dataset size, noise level and noise distribution affect the performance of data sampling techniques, and to identify which perform best when data is noisy and imbalanced. Since the objective is not to identify the best learners, we present results averaged across 11 learners. By selecting learners from various machine learning paradigms, and considering the average performance across them, we can focus on the impact of data sampling, in general, on classification performance. Note that due to space considerations, only a small portion of the experimental results could be presented in this paper.

A. Impact of Class Distribution on Sampling Technique Performance

This section investigates the impact of class distribution on the four sampling techniques. Figure 1 shows the results of using RUS, ROS, SM and WE to alleviate the class imbalance problem using relatively clean data ($NL = 10$, $ND = 0$), but with varied class distributions. The results in Figure 1 are averaged across all 11 learners.

Figure 1a shows the performance of the four sampling techniques (and no sampling, None) for class distributions CD={10,8,6,4,2,1}. At higher class distributions, there is relatively little difference between the four sampling techniques. At CD=10, only WE and SM show (slight) improvement over None, but as class imbalance becomes more severe, the other sampling techniques also improve performance. For CD≤4, all four sampling techniques result in higher AUC values than None. At the most severe levels of imbalance, RUS and SM show the greatest improvement over None, while WE shows the smallest improvement. Although it does not usually result in the highest AUC values, RUS is the most robust to changing the level of imbalance within the data, suffering from the smallest performance loss as class distribution is changed from 10% to 1%.

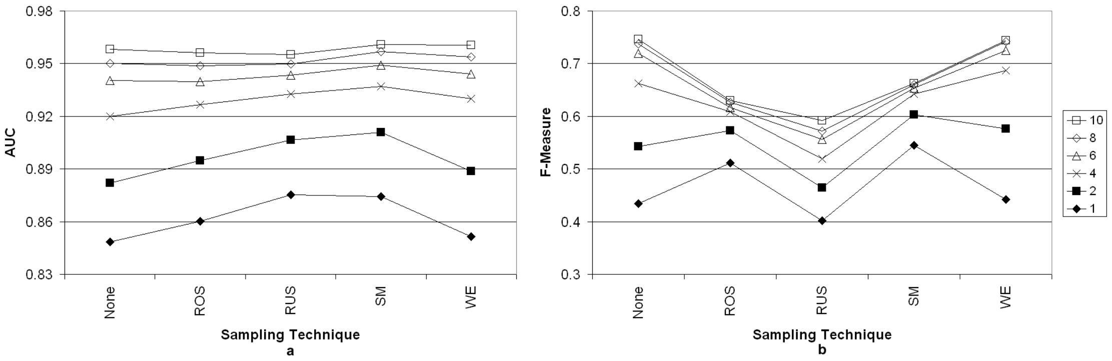

Fig. 1. Performance of sampling techniques at different class distributions, NL=10, ND=0, DS=1500, averaged over all learners and datasets.

Figure 1b shows a much different picture using F-Measure. Three of the four sampling techniques (RUS, ROS and SM) result in lower F-Measures than None using the least imbalanced datasets. Only Wilson's Editing shows (slightly) better performance when CD≥6. As the imbalance in the data becomes more severe, however, oversampling improves classification performance. SM and ROS both outperform None, RUS and WE at these class distributions. RUS, which was among the best sampling techniques for highly skewed data using AUC, results in the worst performance at every class distribution using F-Measure to evaluate the models. Using F-Measure, ROS and SM are relatively robust to changes in class distribution (as the imbalance level increases, the deterioration of the F-Measure for SM and ROS is relatively small).

B. Impact of Noise Level on Sampling Technique Performance

In this section, we compare the performance of the four sampling techniques at difference levels of noise, and evaluate the impact that increasing the amount of class noise within the data has on the performance of each. The results presented in this section are based on the average performance of all 11 learners and all 6 class distributions. Only noise injected using proportional random corruption (PRC), or ND=50, is used so that the class distribution before and after noise in injection is the same. Figure 2 shows the performance of each sampling technique at each noise level.

In Section IV-A, using very clean data, the benefit achieved by the use of sampling was modest. However, using noisy and imbalanced data, the impact of sampling is generally more substantial. Figure 2a shows the performance of the sampling techniques measured using AUC. The two best sampling techniques are RUS and SM, which perform similarly regardless of noise level. WE and ROS also improve performance over None, but not to the degree that RUS and SM do. All four techniques outperform None at all levels of noise. Figure 2a also shows the relative impact of noise on the performance of each sampling technique. Although the lines for each noise level are similar, they are not parallel, indicating that noise does not impact each sampling technique in the same manner. Instead, there is a more pronounced difference between the sampling techniques' performances at higher noise levels. Models built with sampling not only perform better (achieve higher AUCs), but are also more robust to increasing noise.

Relative to the F-Measure in Figure 2b, data sampling is most beneficial when data is both imbalanced and noisy. At the low noise levels, there is relatively little improvement achieved by sampling. At the highest level of noise, however, all four sampling techniques substantially improve classification performance. SM is the best sampling technique regardless of noise level, while RUS generally performs poorly relative to the F-Measure.

C. Impact of Noise Distribution on Sampling Technique Performance

This section investigates the impact of the noise distribution on the performance of these techniques. That is, what type of noise (N→P or P→N) has the most impact on data sampling, and which sampling techniques are most affected by varying the distribution of noise? Figure 3 shows the performance of each sampling techniques (again, averaged across all 11 learners) at five different noise distributions using noise level NL=40. This noise level is used because it is the most impactful (as shown in Section IV-B) level of noise that can be used for all noise distributions. Recall from Section III-B4 that NL=50, ND=100 is not possible because such a dataset would have no minority class examples.

Figure 3a shows the performance of the sampling techniques at various noise distributions using AUC. Note that at lower noise distributions (when most or all noisy examples are labeled majority class examples) there is relatively little difference between the performance of the sampling techniques. Using ND=0 (no mislabeled minority class examples) the two random sampling techniques perform about as well as None, while the two intelligent sampling techniques obtain a slightly higher AUC. As ND is increased, the improvement obtained by sampling increases substantially, and the relative performance of the sampling techniques changes. For ND≥50, RUS results in the highest AUC. Each of the sampling techniques is affected by increasing ND. With the exception of RUS, all sampling techniques perform better at lower levels of ND than at higher levels. In other words, mislabeled minority class examples are more detrimental to learning than mislabeled majority class examples. RUS follows this trend as well, but when ND is increased from 75 to 100, its performance improves slightly. RUS is the least impacted by varying the noise distribution, while Wilson's Editing is the most impacted of the four sampling techniques.

Using F-Measure (Figure 3b), the results are very similar to using AUC. That is, RUS is the least impacted by changing the noise distribution, while WE is the most affected. As is often the case using F-Measure, data sampling can actually have a negative impact on performance. Figure 3 shows that mislabeled majority class examples are less impactful than mislabeled minority class, and that sampling is less beneficial at lower levels of ND (less mislabeled minority examples) than higher levels. Therefore, we conclude both the level of noise in the data and the type of noise determines whether data

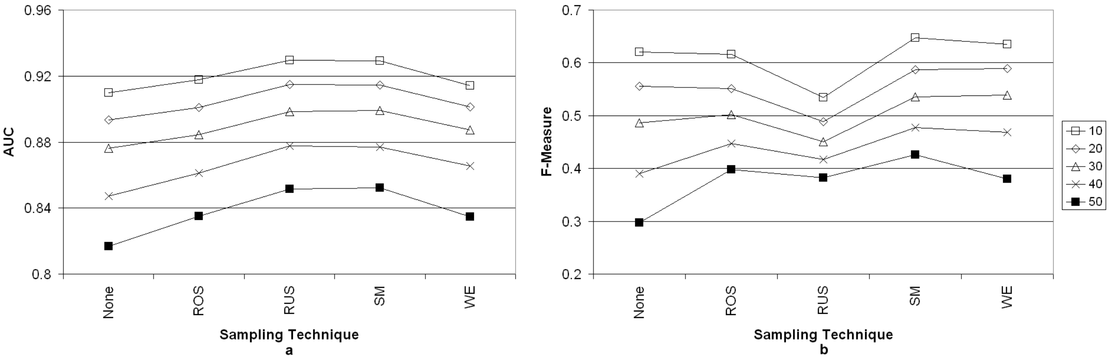

Fig. 2. Performance of sampling techniques at different noise levels, ND=50, DS=1500, averaged over all learners, datasets and class distributions.

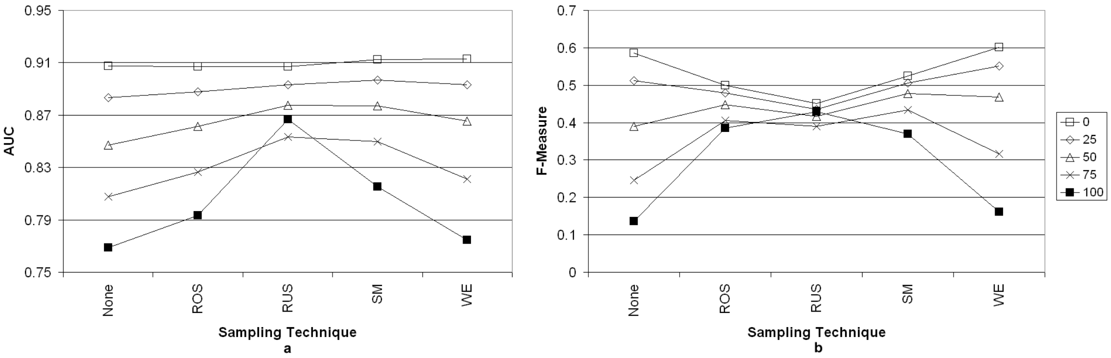

Fig. 3. Performance of sampling techniques at different noise distributions, NL=40, DS=1500, averaged over all learners, datasets and class distributions.

sampling will produce beneficial results. For all ND≥50, all four sampling techniques improve performance. Although the results in previous sections indicate that RUS is not a very good sampling technique when using F-Measure to evaluate classifiers, this section shows that it can be very effective depending on the distribution of noise in the data.

D. Impact of Dataset Size on Sampling Technique Performance

This section examines the impact of dataset size on the performance of the four sampling techniques. As explained in Section III-B1, these experiments use only two of the four datasets (LetterA and OptDigits) because only these two datasets facilitate all experimental parameters at all three sizes. Figure 4 shows the average results across all 11 learners using the least detrimental noise parameters (NL=10, ND=0) and each of the sampling techniques.

Figure 4 shows the impact of changing the dataset size on the performance of the sampling techniques. As expected, models built using smaller datasets result in worse performance than those built on larger datasets. Increasing the size of the datasets from 1500 to 3000 results in a substantial increase in performance, while the improvement achieved by adding an additional 1500 examples (increasing dataset size from 3000 to 4500) is less significant. RUS is the most sensitive sampling technique relative to dataset size. In most cases, sampling does not significantly improve classification performance. Recall from Sections IV-B and IV-C that data sampling is most beneficial when data is both imbalanced and noisy. The results in Figure 4 are based on relatively clean data.

V. Conclusion

This study presents an extensive investigation of data sampling when data is both imbalanced and noisy. We present the performance of four data sampling methods, comparing their performance while varying the size, class distribution, noise level and noise distribution within the training data. Through a comprehensive suite of experiments, we identify which of the sampling techniques result in the greatest performance improvements and which are most impacted by changes in these data quality parameters.

Models trained without sampling, using imbalanced but clean data, with easily learnable concepts, are often as good as or better than models trained using data sampling. However, as noise is injected into the data, the concepts become more difficult to learn and traditional machine learning algorithms have more difficulty distinguishing between examples of the various classes. In these cases, when data is both imbalanced and noisy, data sampling does significantly improve performance. Similarly, even when data is relatively clean, if the level of imbalance is severe enough, data sampling can be beneficial to performance. All four data sampling techniques improved performance on noisy and imbalanced data, but RUS resulted in a lower F-Measure (but often higher AUC) when learning from imbalanced but clean data.

While the differences between performance metrics are not the focus of this paper, our results show that conclusions drawn about the effectiveness of data sampling can vary depending on the performance metric. Using ROC curves, which consider false positive and true positive rates, data sampling typically improves performance, while using F-Measure (which considers true positive rate and precision),

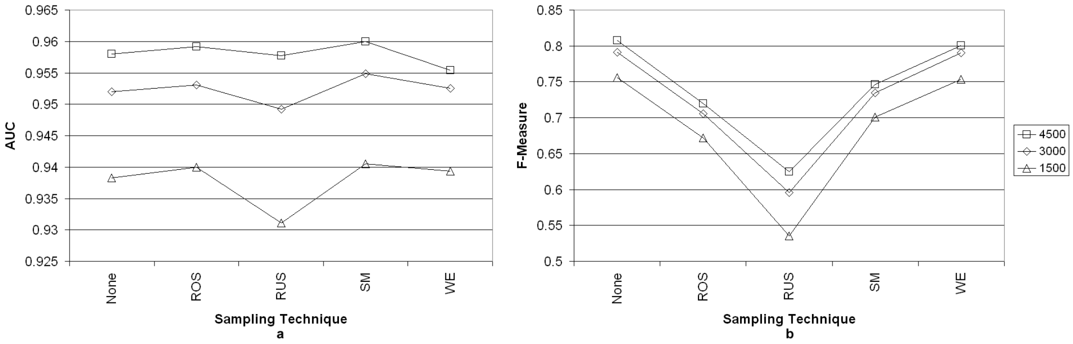

Fig. 4. Performance of sampling techniques at different dataset sizes, NL=10, ND=0, averaged over all learners and class distributions, for datasets LetterA and Optdigits only.

data sampling can be detrimental to learning when the data is relatively easy to learn from.

In general, Wilson's Editing is a relatively safe technique, resulting in a modest improvement regardless of noise level or performance metric. Other techniques, however, are significantly better when data is more severely imbalanced or contains high levels of noise. Both SMOTE and ROS improve the F-Measure when data is severely imbalanced or contains moderate to heavy noise. Using AUC however, SMOTE typically outperforms ROS. Regardless of performance metric, both SMOTE and ROS were similarly affected by changes in the experimental factors. RUS typically performs as well as or better than SMOTE using AUC, but is generally worse than SMOTE relative to the F-Measure. However, when the noise consists largely of mislabeled minority class examples, RUS significantly outperforms the other techniques. With the exception of dataset size, RUS was the least sensitive to variations in the four experimental factors. While it did not always perform the best, its performance changed the least as these factors were varied.

To conclude, data sampling can be a very effective way to improve learner performance, but the performance of data sampling techniques is subject to the availability (dataset size and class distribution) and quality (noise level and noise distribution) of the data. Using AUC, data sampling rarely hurt performance, but only significantly improved performance when data was at least moderately skewed or noisy. Using F-Measure, data sampling often significantly hurt performance when applied to slightly skewed or noisy datasets, but did improve performance when data was either severely noisy or skewed, or contained moderate levels of both noise and imbalance.

Future work can include learning algorithms and sampling techniques not considered here. Additional datasets and methods for simulating noise can also be utilized, for example using datasets derived from a generative model where the distributions are known.

References

[1] X. Zhu and X. Wu, "Class noise vs attribute noise: A quantitative study of their impacts," *Artificial Intelligence Review*, vol. 22, no. 3-4, pp. 177–210, November 2004.

[2] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Skewed class distributions and mislabeled examples," in *Proceedings of the IEEE International Conference on Data Mining - Workshops (ICDMW'07)*, Omaha, NE, USA, October 2007, pp. 477–482.

[3] D. Wilson, "Asymptotic properties of nearest neighbor rules using edited data sets," *IEEE Trans. on Systems, Man and Cybernetics*, no. 2, pp. 408–421, 1972.

[4] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Smote: Synthetic minority oversampling technique," *Journal of Artificial Intelligence Research*, no. 16, pp. 321–357, 2002.

[5] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco, "An empirical study of the classification performance of learners on imbalanced and noisy software quality data," in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2007)*, Las Vegas, NV, March 2007, pp. 651–658.

[6] G. M. Weiss, "Mining with rarity: A unifying framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.

[7] N. Japkowicz and S. Stephan, "The class imbalance problem: a systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–450, 2002.

[8] C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, "Hybrid sampling for imbalanced data," *Journal of Integrated Computer-Aided Engineering*, vol. 16, no. 3, pp. 193–210, 2009.

[9] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, 2003.

[10] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, Corvalis, OR, June 2007, pp. 935–942.

[11] J. Van Hulse and T. M. Khoshgoftaar, "Knowledge discovery from imbalanced and noisy data," *Data and Knowledge Engineering*, vol. 68, no. 12, pp. 1513–1542, December 2009.

[12] D. Gamberger, N. Lavrač, and C. Grošelj, "Experiments with noise filtering in a medical domain," in *Proceedings of the 16th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1999, pp. 143–151.

[13] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *Journal of Artificial Intelligence Research*, vol. 11, pp. 131–167, 1999.

[14] T. M. Khoshgoftaar, S. Zhong, and V. Joshi, "Enhancing software quality estimation using ensemble-classifier based noise filtering," *Intelligent Data Analysis: An International Journal*, vol. 6, no. 1, pp. 3–27, 2005.

[15] J. Van Hulse and T. M. Khoshgoftaar, "Class noise detection using frequent itemsets," *Intelligent Data Analysis: An International Journal*, vol. 10, no. 6, pp. 487–507, December 2006.

[16] C. M. Teng, "Correcting noisy data," in *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999, pp. 239–248.

[17] A. Asuncion and D. Newman, "UCI machine learning repository," *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 2007, university of California, Irvine, School of Information and Computer Sciences.

[18] X. Zhu and X. Wu, "Cost-guided class noise handling for effective cost-sensitive learning," in *4th IEEE International Conference on Data Mining (ICDM 2004)*, November 2004, pp. 297–304.

[19] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco, California: Morgan Kaufmann, 2005.

[20] R. Barandela, R. M. Valdovinos, J. S. Sanchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" *In Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR/SPR'04), Lecture Notes in Computer Science 3138*, no. 806-814, 2004.