**A comparison of statistical methods for improving rare event classification in medicine**

Alyssa Forber1,2, Elizabeth Juarez-Colunga1,2, Susan Calcaterra3, Kathryn Colborn1\*

1Department of Biostatistics and Informatics, Colorado School of Public Health, University of Colorado Denver, Aurora, CO, USA,

2Adult and Child Consortium for Health Outcomes Research and Delivery Science, University of Colorado Denver, Aurora, CO, USA,

3[Susan’s]

\*Corresponding author:

Kathryn Colborn, PhD

13001 E. 17th Place, Room C3011

Aurora, Colorado, USA 80045

Kathryn.colborn@ucdenver.edu

**Abstract**

**Introduction**

Over at least the past decade, there have been numerous articles comparing the performance of various statistical methods for handling the imbalanced learning problem, i.e., when one class is significantly larger than the second class. In many of these studies, various sampling techniques have been compared, such as over-sampling the minority class, under-sampling the majority class, or a sophisticated combination of both.1, 2 As usual, no single method always performs the best when predicting on a hold-out set (CITE). However, all methods achieve the goal of increasing sensitivity (if we assume the minority class is the positive outcome class), which is of course very poor if the imbalance is ignored.

In medicine, clinical prediction or classification models have been popular for several decades, and in contrast to sampling techniques, classification of rare events has typically been handled after the model has been fit using a probability threshold cutoff, typically determined by choosing that which maximizes the area under the curve (AUC) produced by the receiver operating characteristic curve (ROC). The default method for classification from a model that produces a predicted probability for each patient is to round at 0.5. However, when prevalence is substantially less than 0.5, the result is very low sensitivity. The ROC curve displays the sensitivity and specificity tradeoff for various probability threshold values, and without any reason to do otherwise, the statistician will simply choose the threshold that maximizes both simultaneously (i.e., that which maximizes AUC).

One publication in the computer science literature compared sampling techniques to ROC analysis for handling imbalanced data (CITE: Maloof, 2003). In this paper, the author compared the classification performance of various machine learning algorithms when fit to under- and over-sampled training data versus simply moving the decision-threshold for algorithms fit to the full training data. Similar to what Breiman found in 1984 (CITE), sampling or moving the decision threshold had similar effects on classification performance.

In this study, we compare under-sampling, over-sampling, and more recent sampling techniques, such as synthetic minority over-sampling technique (SMOTE),1 with ROC analysis in two clinical classification problems. The first problem is to classify hospitalized patients at risk of developing chronic opioid therapy (CITE: Calcaterra, 2018). In this problem, context matters when choosing the best performing model (i.e., physicians consider the impacts of misclassification on their patients, and might weight sensitivity and specificity differently, depending on the patient). The second problem is to develop an algorithm for identifying postoperative surgical site infections (SSI) using electronic health record data (EHR; CITE: Colborn). In this problem, the goal is to maximize sensitivity and specificity, and there is no impact on patients, as the infections are found after they occur (i.e., it is more of a data mining problem, where we seek to identify what has happened electronically rather than through manual chart review). We also present a simulation study where we compare sampling techniques to ROC analysis for varying rates of disease prevalence. Finally, we conclude with a discussion of the tradeoffs of each approach and how these tradeoffs might affect clinical decisions.

**Methods**

We present two case studies in the next two sections. In the first study, the objective was to classify patients at risk of developing chronic opioid therapy within one year following their initial hospitalization. This event occurred in approximately 5% of patients. In the second study, the objective was to develop an algorithm for identifying postoperative SSIs using EHR data. SSIs occurred in 3.4% of patients. For each problem, the following approach was used to carry out variables selection and classify patients with and without the outcome of interest.

The data were first split into training and test sets, with approximately 2/3 of the data used for the training set and 1/3 held out for testing. A temporal split was used, where the training data consisted of earlier data and the test set were later, according to calendar time. Generalized linear models (GLM) with least absolute shrinkage and selection operator (lasso) penalties were used to classify patients. Ten-fold cross-validation was performed for each method to determine the optimal lambda (i.e., penalty value) for classification and variable selection using the glmnet package3 in R.4 Models were fit to the full training data and subsets of the training data using a) under-sampling, b) over-sampling, and c) SMOTE. For each fit, we tested the classification performance on the hold-out set with respect to various performance statistics using a) a 0.5 threshold and b) the optimal threshold according to Youden’s *J* statistic.5

For under-sampling, we took a random sample of patients from the majority class in order to match the size of the minority class. For over-sampling, we generated exact copies of members of the minority class to create an equal size to that of the majority class. Finally, for SMOTE, we applied 200% over- and under-sampling and k=5 nearest neighbors to generate the synthetic data points. SMOTE was performed using the DMwR package in R.6

**Case Study 1: Classification of patients at risk of developing chronic opioid therapy**

Details of this study have been published previously (CITE: Calcaterra). In this study, we used EHR data from Denver Health patients with an initial hospitalization between 2008 to 2014. We defined chronic opioid therapy as >= 90-day supply of opioids with < 30-day gap in supply over a 180-day period or receipt of >= 10 opioid prescriptions over one year. Of the 27,705 patients, 5% developed COT within a year of their hospital discharge. There were 35 explanatory variables included in model fitting, including: age, race/ethnicity, gender, history of chronic pain, …, and length of hospital stay.

Table 1 summarizes the results from each model fit to the full training data and sampled versions of the training data, and various choices of threshold. As expected, if we use a default threshold of 0.5 on the model fit to the full training data, the sensitivity is very low (8%), yet the specificity and accuracy are quite high (99% and 96%). Using Youden’s *J* to optimize AUC, the threshold was 0.043, with corresponding sensitivity of 85%, specificity of 73%, and AUC equal to 0.73 in the full training data. The models fit to sampled data also showed improvements in sensitivity. Using Youden’s *J* to choose the threshold in the sampled data consistently resulted in higher sensitivity than specificity, lower accuracy, and approximately equal AUC when compared to using 0.5 as the threshold (i.e., the approximate prevalence in sampled data). Interestingly, the threshold was about 10% lower than the prevalence in the under-sampled and over-sampled data; whereas the threshold is almost always equivalent to the sample mean in full training data (Table 2; and other sources: CITE). In these data, it appears that similar results can be achieved through either sampling or using a threshold cutoff or both. The decision of which method is optimal is then left to the user, depending on which statistics he/she prioritizes in the context of the potential for misclassifying patients at risk of developing COT.

**Table 1. Results comparing models fit to the opioid data.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Threshold** | **Specificity** | **Sensitivity** | **NPV** | **PPV** | **Accuracy** | **AUC** | **Covariates** |
| Unsampled | 0.043 | 72.8 | 85.4 | 99.2 | 11.7 | 73.3 | 86.4 | 31 |
| Down Sampled | 0.41 | 74.3 | 83.3 | 99.1 | 12 | 74.6 | 86.4 | 34 |
| Up Sampled | 0.399 | 73.8 | 85.4 | 99.2 | 12.1 | 74.3 | 86.5 | 34 |
| SMOTE | 0.472 | 84.1 | 74.2 | 98.7 | 16.5 | 83.7 | 86.4 | 33 |

**Case Study 2: Identification of surgical site infections**

Details of this study have also been published previously (CITE: Colborn, ICHE). The National Surgery Quality Improvement Program (NSQIP) utilizes trained nurse reviewers to identify postoperative complications using manual chart review and phone calls at more than 680 hospitals throughout the world (CITE). Due to the high cost of the program, nurses can only assess 10-15% of operations at large-volume hospitals. Thus, it is desirable to create an algorithm that can identify postoperative complications in the absence of manual chart review. SSIs are the most common postoperative complication in the NSQIP data. Unfortunately, more than 50% of SSIs occur after the patient has been discharged from the hospital, making it potentially difficult to find them in the EHR unless the patient’s follow-up care was captured by the electronic data source. In this study, we sought to develop a model to identify SSIs using EHR covariate data and NSQIP SSI outcome data from patients who underwent surgery at the University of Colorado Hospital between 2013-2016. SSIs occurred in just 3.4% of 6,840 patients. Covariates we included were binary indicators for antibiotic prescriptions, binary indicators for common procedural terminology (CPT) codes, binary indicators for international classification of disease version 9 (ICD-9) codes, and a continuous measure of CPT-specific SSI event rate for the initial operation (i.e., a measure indicating the risk of the operation, given past experiences of patients from the entire NSQIP dataset of more than 5.4 million patients). There were 136 independent variables total. Results from fitting the lasso models to the full training data and sampled training data are provided in Table 2.

In these data, fitting a model to the full training data and using Youden’s *J* to choose the threshold showed a slight edge over models fit to the sampled data, although all methods performed very similarly (Table 2). Consistent with what was seen in the opioid study, over-sampling and SMOTE select the largest number of predictors.

**Table 2. Results comparing models fit to the surgical data.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Threshold** | **Specificity** | **Sensitivity** | **Accuracy** | **NPV** | **PPV** | **AUC** | **Covariates** |
| Unsampled | 0.036 | 0.9 | 0.8 | 0.9 | 0.99 | 0.24 | 0.89 | 35 |
| Down Sampled | 0.48 | 0.87 | 0.82 | 0.87 | 0.99 | 0.2 | 0.89 | 20 |
| Up Sampled | 0.45 | 0.91 | 0.79 | 0.9 | 0.99 | 0.24 | 0.89 | 123 |
| SMOTE | 0.15 | 0.79 | 0.89 | 0.8 | 0.99 | 0.14 | 0.88 | 88 |

**Simulation study**

We conducted a simulation study to compare the prediction performance of models fit to the full simulated training set versus models fit to sampled versions of the training set across different prevalence rates. We used the chronic opioid therapy dataset by selecting three of some of the most significant predictors—age, chronic pain at discharge, and receipt of opioid at discharge. We first ran a logistic regression with these three variables in order to set our coefficients for the linear predictor. Then we simulated using a logistic distribution to create the new datasets, changing the prevalence by adjusting the intercept. From there, we employed the same methods as in the case studies to get results for sensitivity, specificity, and accuracy with various sampling and cutpoints.

**Table 3. Simulation results from models fit to the full simulated and sampled simulated training set by probability threshold (either 0.5 or that chosen using Youden’s *J*).**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prevalence=3%** | **Threshold** | **Sensitivity** | **Specificity** | **Accuracy** | **AUC** | **Covariates** |
| Full training | 0.03 | 70 | 75 | 74 | 79 | 6 |
| Full training 0.5 | 0.50 | 2 | 100 | 97 | 79 | 6 |
| Under-sampled | 0.49 | 70 | 74 | 74 | 78 | 9 |
| Over-sampled | 0.48 | 70 | 74 | 74 | 78 | 27 |
| SMOTE | 0.41 | 70 | 74 | 74 | 78 | 15 |
|  |  |  |  |  |  |  |
| **Prevalence=5%** |  |  |  |  |  |  |
| Full training | 0.05 | 69 | 74 | 74 | 78 | 7 |
| Full training 0.5 | 0.50 | 4 | 100 | 95 | 78 | 7 |
| Under-sampled | 0.49 | 69 | 74 | 74 | 78 | 9 |
| Over-sampled | 0.49 | 69 | 74 | 74 | 78 | 23 |
| SMOTE | 0.41 | 69 | 74 | 74 | 78 | 16 |
|  |  |  |  |  |  |  |
| **Prevalence=10%** |  |  |  |  |  |  |
| Full training | 0.10 | 68 | 74 | 73 | 77 | 8 |
| Full training 0.5 | 0.50 | 8 | 100 | 91 | 77 | 8 |
| Under-sampled | 0.49 | 68 | 74 | 73 | 77 | 9 |
| Over-sampled | 0.49 | 68 | 74 | 73 | 77 | 18 |
| SMOTE | 0.42 | 68 | 73 | 73 | 77 | 19 |
| **Prevalence=20%** |  |  |  |  |  |  |
| Full training | 0.20 | 66 | 73 | 72 | 76 | 8 |
| Full training 0.5 | 0.50 | 21 | 97 | 82 | 76 | 8 |
| Under-sampled | 0.49 | 66 | 73 | 72 | 76 | 9 |
| Over-sampled | 0.49 | 66 | 73 | 72 | 76 | 13 |
| SMOTE | 0.42 | 66 | 72 | 71 | 75 | 23 |
| **Prevalence=40%** |  |  |  |  |  |  |
| Full training | 0.39 | 0.66 | 0.71 | 0.69 | 0.74 | 9 |
| Full training 0.5 | 0.50 | 0.49 | 0.84 | 0.70 | 0.74 | 9 |
| Under-sampled | 0.49 | 0.66 | 0.71 | 0.69 | 0.74 | 9 |
| Over-sampled | 0.49 | 0.65 | 0.71 | 0.69 | 0.74 | 10 |
| SMOTE | 0.41 | 0.64 | 0.71 | 0.68 | 0.73 | 28 |
|  |  |  |  |  |  |  |
| **Prevalence=50%** |  |  |  |  |  |  |
| Full training | 0.49 | 63 | 72 | 68 | 73 | 9 |
| Full training 0.5 | 0.50 | 63 | 72 | 68 | 73 | 9 |

At every prevalence rate, the ideal cut-point based off of the Youden Index was equal to the prevalence—e.g. 5% outcome, 0.05 cut-point.

^ that’s really the only big thing we see in the simulation, other than that all the results are very, very close and there is no real difference between sampling when you are using the Youden index.

**Discussion**

Discuss the findings from each and how they compare or contrast and what we learned from the simulation study. Conclude with some remarks about how we’ve shown a process for handling imbalance and that context matters, so there is no recipe for the best approach – one might want to follow a process similar to what we did.

* SMOTE cutoff for surgical infections was really low
* All cutoffs for sampling were all under 0.5 in the 0.4 range
* With opioid, the SMOTE sensitivity was a lowest and specificity the highest, but the opposite was true for surgical infections where sensitivity was the highest and specificity the lowest
* Accuracy highest for SMOTE in opioid and lowest for surgical infections
* Fewest covariates for unsampled with opioid, and highest with up and down, but with surgical, down was the lowest number of covariates, though unsampled was second lowest
* All methods really performed pretty similarly, and knowing that the ROC curves are so similar, it seems like sampling doesn’t make enough of a difference, especially since you could specify wanted a minimum sensitivity or specificity if you needed to in a certain setting
* The similarity of the methods is really evident in the simulation as well, there is hardly a detectable difference between each of them, so it seems going through the trouble of sampling is not worth it?
* The sim results show the cutoff to be exactly the prevalence, but with the opioid it is slightly under and with surgical it is slightly over

**References**

1. Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 2002; 16: 321-357.

2. Japkowicz NS, Shaju. The class imbalance problem: A systematic study. *Intelligent Data Analysis* 2002; 6: 429-449.

3. Jerome Friedman TH, Robert Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* 2010; 33: 1-22.

4. R Core Team. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing* 2015.

5. Youden WJ. Index for rating diagnostic tests. *Cancer* 1950; 3: 32-35.

6. Torgo L. Data Mining with R, learning with case studies. *Chapman and Hall/CRC* 2010.