Proceedings of the 2015 INFORMS Workshop on Data Mining and Analytics
M. G. Baydogan, S. Huang, A. Oztekin, eds.

DEVELOPING A REBALANCING PARAMETER TABLE FOR BINARY CLASSIFICATION MODELING

Matthew A. Lanham
Virginia Polytechnic Institute and State University
Department of Business Information Technology (0235)
1007 Pamplin Hall, Blacksburg, VA 24061
lanham@vt.edu

Ralph D. Badinelli
Virginia Polytechnic Institute and State University
Department of Business Information Technology (0235)
1007 Pamplin Hall, Blacksburg, VA 24061
ralphb@vt.edu

Abstract

We investigate rebalancing approaches for training binary classification algorithms and identify certain combinations of factors that could be strategically implemented in a predictive modeling engine so as to improve analytical performance over time when decision-support time constraints must be considered. It is known that rebalancing the training data will often lead to improved predictive performance on out-of-sample testing data. However, in practice a firm will often need to automatically train and test thousands of models for different product segments which can take a considerable amount of time. To do this accurately and efficiently, the modeler would likely need to know for each modeling set having its own respective level of minority class imbalance which method performs the best (e.g. based on AUC) for different types of algorithms, as well as the expected runtime for each case based on the size of the dataset being used. Using category sales data from a national retailer we examine some rebalance results for different algorithms to motivate the need for these parameters to be saved and incorporated into a decision model that would help a retailer identify which rebalance algorithm-binary classification model combination to focus on when known decision-support delivery time is specified.

Keywords: Class Imbalance, Binary Classification, Life Cycle Management

Introduction

Binary classification problems arise frequently in business situations where a firm is employing predictive analytics. For example, a firm may want to predict the probability that a customer will leave and go to a competitor (churn), know if a product will sell or not sell (merchandising), customer will default on their credit card balance (finance), or if a customer will respond to an advertisement (marketing campaign). It has been observed that class imbalance is common to all of these problems in practice and can have negative impacts on modeling performance.

Firms will employ binary classification algorithms on historic observations to generate a classifier that is as accurate as possible so that when new measurements arrive in the future, they can be fed into the model to gauge which class the record would most likely correspond to. Our primary research question is which rebalance technique and algorithm type is most appropriate to employ when a firm is constrained with a finite amount of computing resources and must deliver decision-support predictions on time? This question is motivated by the frequently occurring scenario of having to build many binary classification models on datasets that have similar data characteristics yet varying degrees of imbalance.

Ignoring the impact that an unbalanced data set can have on model estimation and the model's accuracy can be severe when the classes are not perfectly separable (Hand & Vinciotti, 2003) or complexity is high (Japkowicz & Stephen, 2002). It has been shown by many investigators that when the response is heavily imbalanced these characteristics of the data can have significant effects on an algorithms ability to learn (Japkowicz & Stephen, 2002). The reason for this is because learning algorithms tend to focus on the response class that has the greatest majority, thus in rare event modeling the difficulty increases. When the proportion of the response class is skewed, evaluation of the error or accuracy of a classier is at risk due to the insufficiency of the minority class. The reason for misclassification is in part due the choice of the method employed as well as the unique features in the dataset.

There have been advances in binary classification research but Menardi and Torelli (2014) state that, *"the research community is still pursuing an undisguised and unified approach to the class imbalance problem."* Moreover, each contribution on the class imbalance problem shows how one approach outperforms existing methods in some aspects, but is itself outperformed by other methods in other important aspects. It is often not clear when one technique would be preferred over another. A literature review of the remedies for unbalanced data can be found here (He & Garcia, 2009; Kotsiantis, Kanellopoulos, & Pintelas, 2006).

To briefly summarize some of these findings, in logistic regression modeling when classes are unbalanced the conditional probabilities of the minority class are underestimated (Cramer, 1999; Owen, 2007). Linear Discriminate Analysis (LDA) assumes the there is an equal covariance matrix of the two classes. The common covariance is a weighted average of the two class matrices, so if one class is heavily favored over another more bias will exist (Hand & Vinciotti, 2003). Xie and Qiu (2007) and Xue and Titterington (2008) provide an interesting debate on this issue. Nonparametric approaches are more flexible than these traditional parametric approaches, because they build classification rules that as a whole create the classifier. The rules are created based on optimizing an objective function, which can be a problem when the objective function criterion is to maximize (minimize) global accuracy (error), the classifier will tend to perform well on the majority class and not the minority. Classification trees for example tend to construct less complex models having high accuracy for the majority class but low accuracy for the minority class, and when complex models are constructed are likely to yield over fit models (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Cieslak & Chawla, 2008). Similarly, for k-Nearest-Neighbors (kNN) having a fixed class has been shown that as the number of negative responses grows in a data set the likelihood that the nearest neighbor is negative increases as well (Kubat & Matwin, 1997). More sophisticated techniques such as Artificial Neural Networks (ANNs) for binary classification modeling have their theoretical assumptions violated in the presences of unbalanced classes, thus must be corrected when training (Lawrence et al.). For Support Vector Machines (SVMs), an imbalance among classes leads to an inept tradeoff to simultaneously minimize error and maximize margin (Akbani, Kwek, & Japkowicz, 2004).

This short review motivates the fact that training datasets must be rebalanced to achieve robust predictive models, but in truth each potential rebalance technique leads to different predictive performance for different algorithms, as well as requires different amounts of computing time to accomplish.

We structure this short paper by identifying the business problem and framing it into an analytics problem, describe the research design methodology we employ, discuss our motivating results, and finally provide a decision model that incorporates the table of parameters to provide the best possible decision-support based on operational constraints.

Business Problem to Analytics Problem

Retailers must regularly estimate demand for their products to help plan that the right products are provided in the right locations at the right time. Binary classification models are one approach that might be appropriate for certain product types, such as those having low turn and long planning horizons (e.g. service parts). It is common practice for a retailer to segment their customers or products in some fashion (geography, planograms, categories, etc.) to increase the predictive accuracy of their predictive models. The set of potential stock-keeping units SKUs that a retailer may choose from can be as many as 500,000 or more. A traditional merchandise store (e.g. Big K) would carry approximately 16,000 unique SKUs per store with an additional 6,000 seasonal products, depending on the time of year (Cox, 2011). When one multiplies this by the number of stores a chain retailer has (e.g. Walmart has 5,187 stores, Target has 2,155 stores), the number of models required to generate a customized assortment mix of products for any given store can be very complex.

If the models required happen to be binary classification based as focused on in this paper, the retailer would likely need to know for each modeling set having its own respective level of minority class imbalance, which method performs the best (e.g. based on AUC) for different types of algorithms in order to achieve the best predictive fit for each subset. Moreover, how long do each of these scenarios take to generate? However, instead of determining this information for each period’s forecasting run this knowledge could be stored and used as modeling process parameters. We posit this must be performed because the amount of time to build new models at scale can be significant even using the most expensive software packages. Also, those retailers not having the ability or managerial support to take advantage of cloud-based Map Reduce frameworks, such as Hadoop or Spark, must strategically optimize their predictive modeling improvement efforts so as to provide accurate and timely decision-support.

Methodology and Results

Using category sales data from a national retailer where the objective was to estimate the propensity to sell, we examined the results from three randomly chosen product categories. Each category had a total number of records that fit into each of these bins (<100k, 100-200k, 200-200k), and each had different minority class percentages (5-10%, 10-20%, and 20-30%). The datasets were randomly partitioned into 70/30 training-testing datasets, and then the training data was rebalanced using four different rebalancing techniques (up-Sampling, down-Sampling, SMOTE, and ROSE).

The up-Sampling approach resamples the minority class with replacement until the same number of records exists for the minority class as the majority class. Similarly, down-sampling is when the majority class is sampled so as to yield the same number of training records as the minority class. The Synthetic Minority Over-sampling Technique (SMOTE) approach over-samples the minority class by creating "synthetic" examples instead of resampling with replacement. The Random OverSampling Examples (ROSE) is a systematic framework for fixing learning issues that arise from unbalanced data by using a smoothed bootstrap form of resampling (Menardi & Torelli, 2014). Lastly, each rebalanced training set was fed into four different binary classification algorithms (logistic regression, C5.0 decision tree, linear discriminant analysis, Classification tree) to capture the runtime of each of these scenarios.

Based on the samples we used the SMOTE rebalance technique revealed the longest amount of computation time as shown in Table 1. Interestingly, it also leads to a training data set that is not completely balanced.

Num of Obs	**Method**	**Rebalance time (minutes)**	**Minority Class % Post**	**Number training Obs post rebalance**
<100k	up	0.016	0.500	34,908
<100k	down	0.005	0.500	5,694
<100k	smote	1.168	0.429	19,929
<100k	rose	0.051	0.493	20,301
100-200k	up	0.051	0.500	188,048
100-200k	down	0.011	0.500	44,944
100-200k	smote	19.769	0.429	157,304
100-200k	rose	0.347	0.498	116,496
200-300k	up	0.034	0.500	338,000
200-300k	down	0.009	0.500	182,000
200-300k	smote	37.561	0.430	260,000
200-300k	rose	2.331	0.490	260,000

Table 1: Rebalance run time per data set size and rebalance algorithm

Each rebalance approach led to different sized training data sets. Up-sampling for example always led to the largest data set, which in turn resulted in longer model training time for all algorithms tested.

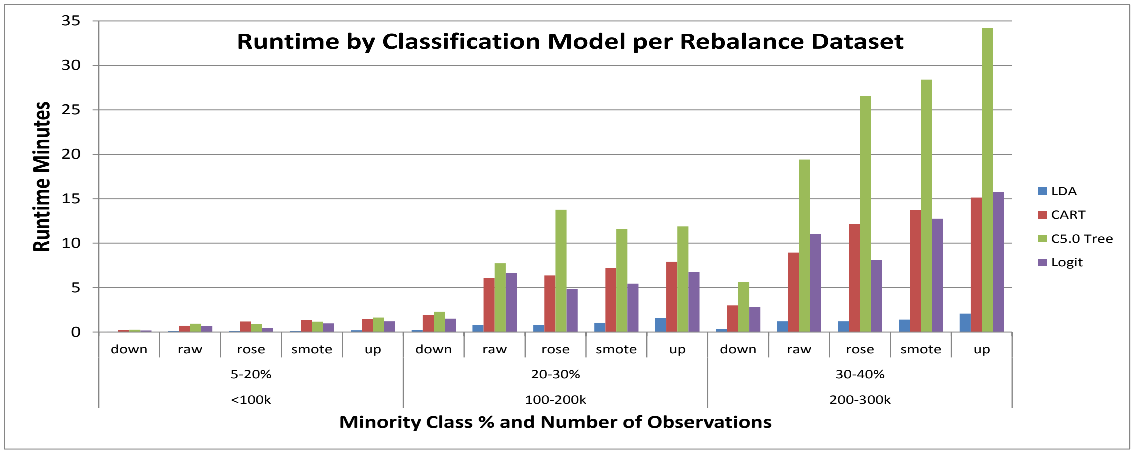

Figure 1: Runtime by classification model for different size datasets and minority class imbalance percentages

A retailer could identify which rebalance algorithm works optimally for each binary classification algorithm and save these results in a parameters table to be used for future scheduled forecasting endeavors. If they cannot perform this initially for all combinations they could randomly choose a subset having different levels of imbalance and algorithms and use the results from that run as their parameters as we performed in this paper.

After having this parameter table they could easily implement the following integer programming model to decide which rebalance methodology to employ for which algorithms where they only need to specify **T**, which is the total amount of time that can be allocated for model re-estimation for the next periods forecasting run. This decision model takes into account the various sized modeling segments, their respective out-of-sample testing data accuracies for each algorithm and rebalance technique, as well as the expected run time for each combination.

Terms and definitions:

x_{ij} = which rebalance algorithm j to use for modeling segment i, $x_{ij} \in \{0,1\};\ i = 1,..,N;$
$j = 1,..,J$

y_{ijk} = which classification algorithm k to refit with the most recent data for modeling segment i for rebalance algorithm j, $y_{ijk} \in \{0,1\};\ i = 1,..,N;\ j = 1,..,J;\ k = 1,..,K$

μ_i = previous percentage of minority class unbalance for modeling segment $i;\ i = 1,..,N$

η_i = total number of current records for modeling segment $i;\ i = 1,..,N$

θ_{ij} = rebalance runtime for modeling segment i and rebalance algorithm j

τ_{ijk} = the training runtime for modeling segment i and rebalance algorithm j and classification algorithm k

α_{ijk} = the testing set accuracy for modeling segment i and rebalance algorithm j and classification algorithm k

Objective function:

$$max \sum_k \sum_j \sum_i \eta_i * \alpha_{i_{ijk}} * y_{ijk}$$

Constraints:

(1) $\sum_{j=1}^{J} x_{ij} = 1,\ \forall\, i$ (only one rebalance algorithm will be chosen per modeling segment)

(2) $\sum_j \sum_i (\theta_{ij} * x_{ij}) + \sum_i \sum_j \sum_k (\tau_{ijk} * y_{ijk}) \leq \boldsymbol{T}$ (total rebalance and model train time must be feasible)

(3) $\sum_i \sum_k y_{ijk} \geq 1,\ \forall\, i$ (each modeling segment needs at least one rebalance algorithm and one binary classification algorithm)

(4) $\sum_j y_{ijk} = 1, \ \forall\, i,k$ (only one rebalance algorithm k for each classification model and segment)

(5) $\sum_k y_{ijk} \leq K x_{ij}, \ \forall\, i,j$ (if we use rebalance algorithm j for any binary class algorithm k for segment i then it must be 1)

(6) $x_{ij} \leq \sum_k y_{ijk}, \ \forall\, i,j$ (if we do not use rebalance algorithm j for any binary class algorithm k for segment i then it must be 0)

(7) $x_{ij}, \, y_{ijk}$ $binary$

Conclusions & Future Research

Unbalanced data sets occur regularly in practice and must be appropriately adjusted when training binary classification models. While an overall framework or methodology still does not exists that works the best for all datasets having various degrees of imbalance, a retailer can strategically test their segments and store those results in a parameters table. Such a table of parameters can be strategically taken advantage of in a decision model to help them identify which rebalance technique to use for each algorithm and modeling segment so that each period’s analytical decision-support activities will lead to the greatest expected model performance gains while satisfying the time constraints imposed on the team and their resources performing the work.

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets *Machine Learning: ECML 2004* (pp. 39-50): Springer.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 321-357.

Cieslak, D. A., & Chawla, N. V. (2008). Learning decision trees for unbalanced data *Machine Learning and Knowledge Discovery in Databases* (pp. 241-256): Springer.

Cox, E. (2011). *Retail Analytics: The Secret Weapon*: Wiley.

Cramer, J. S. (1999). Predictive performance of the binary logit model in unbalanced samples. *Journal of the Royal Statistical Society: Series D (The Statistician), 48*(1), 85-94.

Hand, D. J., & Vinciotti, V. (2003). Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern recognition letters, 24*(9), 1555-1562.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on, 21*(9), 1263-1284.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis, 6*(5), 429-449.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering, 30*(1), 25-36.

Kubat, M., & Matwin, S. (1997). *Addressing the curse of imbalanced training sets: one-sided selection.* Paper presented at the ICML.

Lawrence, S., Burns, I., Back, A., Tsoi, A. C., Giles, C. L., & Chung, A. To appear in: Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys, edited by G. Orr and K.-R. Mwuller and R. Caruana, Springer Verlag, pp. 299-314, 1998. Neural Network Classification and Prior Class Probabilities.

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery, 28*(1), 92-122.

Owen, A. B. (2007). Infinitely imbalanced logistic regression. *The Journal of Machine Learning Research, 8*, 761-773.

Xie, J., & Qiu, Z. (2007). The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern recognition, 40*(2), 557-562.

Xue, J.-H., & Titterington, D. M. (2008). Do unbalanced data have a negative effect on LDA? *Pattern recognition, 41*(5), 1558-1571.