2013

A balanced approach to the multi-class imbalance problem

Lawrence Mosley
Iowa State University

A balanced approach to the multi-class imbalance problem

by

Lawrence Se'kou Denu Mosley

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:

Sigurdur Ólafsson, Major Professor

Dianne Cook

Stephen Gilbert

Heike Hofmann

Lihzi Wang

Iowa State University

Ames, Iowa

2013

DEDICATION

To family, because, *if two lie together then they have heat: but how can one be warm alone?*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The multi-class class-imbalance problem is a subset of supervised machine learning tasks where the classification variable of interest consists of three or more categories with unequal sample sizes. In the fields of manufacturing and business, common machine learning classification tasks such as failure mode, fraud, and threat detection often exhibit class imbalance due to the infrequent occurrence of one or more event states. Though machine learning as a discipline is well established, the study of class imbalance with respect to multi-class learning does not yet have the same deep, rich history. In its current state, the class imbalance literature leverages the use of biased sampling and increasing model complexity to improve predictive performance, and while some have made advances, there are still no standard model evaluation criteria for which to compare their performance. In the presence of substantial multi-class distributional skew, of the model evaluation criteria that can scale beyond the binary case, many become invalid due to their over-emphasis on the majority class observations.

Going a step further, many of the evaluation criteria utilized in practice vary significantly across the class imbalance literature and so far no single measure has been able to galvanize consensus due not only to implementation complexity, but the existence of undesirable properties. Therefore, the focus of this research is to introduce a new performance measure, Class Balance Accuracy, designed specifically for model validation in the presence of multi-class imbalance. This paper begins with the statement of definition for Class Balance Accuracy and provides an intuitive proof for its interpretation as a simultaneous lower bound for the average per class recall and average per class precision. Results from comparison studies show that models chosen by maximizing the training class balance accuracy consistently yield both high overall accuracy and per class recall on the test sets compared to the models chosen by other criteria. Simulation studies were then conducted to highlight specific scenarios where the use of class balance accuracy outperforms model selection based on regular accuracy. The measure is

then invoked in two novel applications, one as the maximization criteria in the instance selection biased sampling technique and the other as a model selection tool in a multiple classifier system prediction algorithm. In the case of instance selection, the use of class balance accuracy shows improvement over traditional accuracy in scenarios of multi-class class-imbalance data sets with low separability between the majority and minority classes. Likewise, the use of CBA in the multiple classifier system resulted in improved predictions over state of the art methods such as adaBoost for some of the U.C.I. machine learning repository test data sets. The paper then concludes with a discussion of the **climbR** package, a repository of functions designed to aid in the model evaluation and prediction of class imbalance machine learning problems.

CHAPTER 1. INTRODUCTION

An introduction of data mining and its applications will be discussed as a build up towards our specific area of investigation. At that junction, research questions of interest will be established and a brief outline of the thesis structure will be given upon the chapter's conclusion.

1.1 Data Mining and the Operations Researcher

Operations research as a discipline was built around the idea that analytical reasoning is the ideal method for evaluating alternatives. The process of selecting one alternative over another involves framing the problem as a highly structured mathematical program where the objective, decision variables, and constraints are made explicit and arranged in a manner that facilitates the search for optimal solutions. With this approach, agencies have been able to minimize costs, determine the best chemical proportions for gasoline blends, create the most efficient schedules, and find feasible fleet assignments across tens of thousands of variables and constraints (Rajgopal, 2001). The systematic organization of classical efficiency problems into solvable frameworks has been so successful that there is a common tautology now that emphatically states "everything is an optimization problem". As subscribers, we have no qualms about this statement's truth. Regardless of our ability, or inability to solve these mathematical programs, model formulation is only possible after a clear understanding of the objectives, inputs, and constraints. As a testament to its importance, industrial engineers have dedicated an entire step in the operations research work flow for this phase alone (Rajgopal, 2001). The "Data Collection" step in the operations research approach is designated as one such point in the work flow where relevant information is to be collected about the system, process, or event of interest. Data collected on subjects of interest serve to characterize the inputs with the hope

that later analysis will disclose important relationships between the characteristics. Questions arise, *Are certain geographical locations more prone to flight delays? Does the orientation of the airport and the subsequent wind drift direction affect arrival times?* Data driven solutions to these questions are used to form the basis for not only the decision variables and constraints of a program, but the inclusion or exclusion of parameters in the objective function, the key differential for solution discrimination. Therefore beyond simply collecting data for record keeping, there is a need to glean applicable knowledge from this information for the formulation of optimization problems.

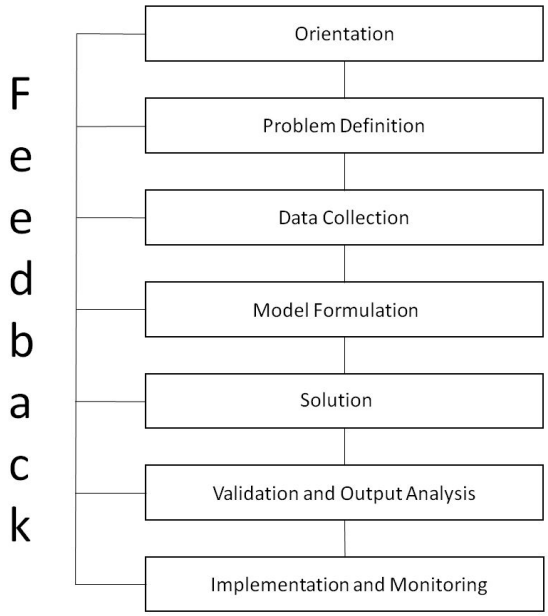

Figure 1.1 The Maynard's Industrial Engineering Handbook visual representation of the operations research approach. This work flow diagram describes the steps necessary for systematic decision making and problem solving.

Sparked by human curiosity and made possible through human ingenuity, the data collection process has become streamlined in such a way that it is now possible to record information across many subjects simultaneously and efficiently which increases the breadth of data. The volume of stored datum combined with the speed in which it is collected and the variety of in-

formation sources form the backbone of what industry has labeled "big data" (Stapleton, 2011). It becomes immediately apparent that to gain guiding insight from such large, high velocity data sets, the information contained must be processed in some automated fashion. It was this same desire for automation that inspired the machine learning scholars of yesteryear to develop the field of data mining to address the "big data" problems of their day (He, 2009). With an alternative paradigm to data analysis than traditional statistical thinking, data mining was introduced as a knowledge discovery tool that could, at the least, semi-automate the process of discovering previously unknown patterns in the data without an a priori hypothesis (Olafsson, 2008). The solutions to this insight search process are unknown patterns which can manifest themselves in two forms: as structured groupings of observations or relationships between input, output data fields. Standard nomenclature denotes the search for natural groupings as unsupervised learning tasks, whereas the investigation into relationships between explanatory variables and a labeled qualitative response is called supervised learning. Returning back to the operations research approach, given a domain context, the successful completion of these tasks can grant the industrial engineer valuable discernment into the model formulation. For example, we may find that *analysis suggests that both geographical location and runway orientation are related to traffic delays; therefore, these effects should be accounted for our formulation through the constraints, decision variables or objective function.* Discussions of data mining thus far have revolved around its use in conjunction with the data collection step to aid industrial engineers in operations research tasks; however the applications of data mining can't be constricted to one field. From the author's own consulting experience data mining techniques have been sought after to differentiate between human and machine generated computer code, group graduate students according to post-baccalaureate school satisfaction, and analyze online text reviews of hotels for specific areas of competitive advantage.

1.2 A Gentle Introduction to Data Mining

Beyond general applications, for the purpose of this thesis, a more in depth discussion of data mining is warranted. As mentioned previously, data mining tasks exist in two realms, where

the learning process is either supervised and unsupervised. Unsupervised learning tasks aim to gather observations into clusters, where the ideal outcome involves the formulation of groups of observations with similar characteristics. In this scenario, the machine learning algorithm uses the input data and subsequently the underlying data structure to determine the optimal cluster membership for each case. This specific type of learning process can also be viewed as a search for latent variables within the data structure, where both the location and number of groups are unknown. Despite having objective data to guide the learning process, there is no way to verify that the clusters drafted by the algorithm are indeed veritable, which lends credence to statement that these techniques are "learning without a teacher" (Tibshirani and Freedman, 2009). That said, practitioners often calculate measures which describe the cluster's compactness and separability; two intuitive measures that quantify the within cluster distances between observations and the between cluster distances, respectfully (Grira, 2005). Based on these measures, a successful unsupervised learning task will create clusters that minimize the within-cluster distances while simultaneously maximizing the between-cluster distances, resulting in clusters that are tightly knit and spread apart. Popular algorithms include the distance based methods like k-means and hierarchical clustering, and self-organizing maps which were derived from the theory of topological maps (Tibshirani and Freedman, 2009). Common applications of unsupervised learning involve market segmentation of customers, grouping of countries with similar to economic output, and signal categorization.

Supervised learning differs from unsupervised learning because well-defined class labels exist for each observation. Given a set of characteristics for the observations, amongst them the corresponding class label, classification models sift through the noise in the data set and output relevant relationships between the characteristics and the class labels. Some of these relationships can be expressed as intuitive patterns, like *"after 5 pm the risk of a network log-in being malicious increases two fold" or "ip addresses that attempt to log-in more than 10 times at perfectly space intervals are 75% more likely to be machines compromised with a Trojan virus than other client terminals"*. To attain these rules, the first step is to partition the data into training and test sets that contain 66.6% and 33.3% of the data, respectfully. With the training set, a model is learned and classification rules are created. These newly developed

classification rules are applied to the test data to assess the accuracy and robustness of the model on observations outside of the original training set. This emulates model usage in the real world. To determine the model's level of accuracy, for each observation in the test set, predictions are derived from the model rule set and are compared with its true class observed from the data. The sum of the number of observations whose predicted class and observed class match are divided by the total number of observations in the data set, which results in high predictive accuracy for models that can recall more of the original observed class labels. This ability to assess the model, as a consequence of having known class memberships, supervises the learning process. As the more formalized branch of machine learning, its uses are pervasive in all branches of science with broad, diverse applications too numerous to list. A survey of applications can be found in Tibshirani and Freedman's *"Elements of Statistical Learning"*.

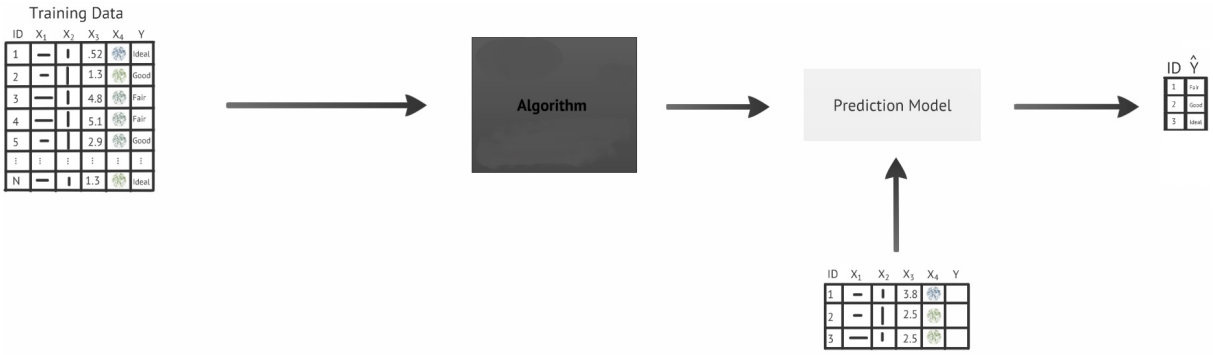

Figure 1.2 A process flow diagram of the supervised learning sequence. After the data collection step, training data is used in conjunction with a machine learning algorithm to create a prediction model. This model can now be used to forecast class memberships of new, previously unforeseen observations.

1.3 Supervised Learning in the Presence of Class Imbalance

Supervised learning models are widely applicable and can offer substantial insights into how the explanatory variables are related to the categorical response variable. The ability of

models to discover useful patterns in data rely on some key assumptions that provide justification for the use of statistical machine learning methods. One such assumption is that there is an underlying deterministic mechanism that generates differences between the groups. This assumption clearly disallows the possibility that the class labels are a sole result of chance. Another fundamental assumption, and the main topic of this thesis, is the requirement that all levels of the categorical outcome variable be evenly distributed. Deviations from this assumption are exhibited when the one of more levels of the response variable are not represented at the same relative frequency as the other levels. This scenario is aptly named, the class imbalance problem (Japkowicz, 2000). Since all classification models seek to find boundaries between classes, in cases where there is a departure from this assumption, meaningful boundaries are hard to ascertain. Reduced to its core, class imbalance makes the very act of prediction more difficult because of the added challenge to group delineation and demarcation (Longadge et. al, 2013). Aside from the added difficulty of partitioning the data space, when the target variable has skewed class distributions, the fundamental intuition behind performance accounting is attacked as imbalance increases. In these situations, performing assessment begins to transition from straight forward ratio calculations and branches into the realm of information theory and matrix reduction. Classifiers are implicitly or explicitly designed to segment the classes to optimize the total number of correctly specified observations. When the objective is merely to maximize the number of observations, classifiers manifest a myopic view of the task which guides them toward the prediction of classes that are over represented in the data set (Kumar and Sheshadri, 2012). As an example, given ninety-eight observations with "positive" labels, a single "negative" observation, and a single "neutral" labeled instance, if the latter two points are not conspicuously separated in the data space then most classifiers would be well suited to create a rule that classifies all observations as a positive group member. The learning rule would achieve ninety-eight percent accuracy, but effectively provide no new knowledge if the initial objective was to gain insight and demarcate boundary lines between the three classes. While the value added of this classification model would be nil, our evaluation criteria returns a value that suggests directly the opposite. In effect, when information about each class is integral, class imbalance severely hinders the effectiveness of traditional accuracy as a performance

measure. This dissertation acknowledges these short comings and seeks to address the class imbalance problem by introducing a novel alternative measure for model evaluation, one that balances the precision and recall metrics across each class.

1.4 Thesis Structure

In this chapter we have discussed how data mining serves as a central part of the modern operations research work flow. An overview of supervising and unsupervised learning was presented, as well as an introduction to the class imbalance problem along with a discussion of its effect on model evaluation. The remainder of this PhD thesis will be outlined in the following sequence.

In chapter 2, we will review the background and literature relevant to the class imbalance problem. It will begin with a formalization of supervised learning tasks within the context of class imbalance. The class imbalance problem will be revisited, including an in-depth discussion of its effects and current approaches. The chapter will conclude with supplemental discussions on material relevant to work done in subsequent chapters. Chapter 3 will begin with a formal proposal of the Class Balance Accuracy measure. Sections will be devoted to its definition, proofs, properties, intuition, and a concluding comparison study highlighting its use as a model evaluation criterion in practice. Where relevant, simulation studies will be discussed to provide additional experimental evidence in support of the measure. In Chapters 4 and 5, two novel applications of class balance accuracy are introduced. In both, Class Balance Accuracy is used within the objective function, where in one the goal is to determine the selection of subsets for instance selection and in the other to determine suitable class experts for a multiple classifier system. Simulation studies for each method are conducted to show how the use of our proposed measure can improve accuracy in the presence of non-separable multi-class data sets. Afterward, Chapter 6 will walk through a software implementation of the methods and procedures introduced to address the class imbalance problem. The chapter will provide interactive documentation for the use of functions specifically designed to assist with model prediction and evaluation in the presence of class imbalance. In conclusion, the final chapter will summarize the key points of this PhD research and discuss future extensions.

CHAPTER 2. REVIEW OF LITERATURE AND BACKGROUND

It is the intent of this chapter is to give the reader a sufficient background understanding of the class imbalance problem and knowledge of the current state of the art.

2.1 Supervised Learning

2.1.1 Introduction

Let a **m**-dimensional vector of measurements be denoted by **X**, where each dimension is identified as x_j such that j = 1,...,m. In conjunction, let **Y** be a singleton element from the set G, that contains, **k**, elements distinguished as g_1,g_2,g_j,...g_k. Combined, the 2-tuple {**X**,**Y**} form one complete data observation. A **n**-dimensional collection of these data observations, $\{\mathbf{X}, \mathbf{Y}\}^n$, form the complete *dataset* from which classification models are trained.

The supervised learning process consists of a structured search through the data space by a chosen member a subset of models within, **M**, the superset of all available models. For any given model, say M_l, when trained with some randomly selected subset of data, classification boundaries are drawn based on the location of optimal separations that maximize some algorithm based measure of separation (Tibshirani and Freedman, 2009). Commonly used assessments of separation are Kullback-Leibler divergence and the Gini coefficient. Boundaries derived from these models may consist of rule based criteria that delineate the classes according to the values of the input variables or archetypal observations that serve as threshold limits where all observations more extreme are deemed to be from another group extant in the data. Due to the diversity of algorithm approaches for boundary detection, there can be a reasonable expectation for a commensurate amount of heterogeneity in model interpretability. Generally, as models increase in complexity, the effects of the explanatory variables are no longer extri-

cable due to the lack of closed form partitionable formulas. Ideally, to preserve the individual input variable effects, their contribution should be structured in such a manner that facilitates easy differentiation, not unlike the concept of partitioning sum of squares in the linear model framework (Kutner et.al, 2004). Interpretability aside, these boundaries established by the models are used for the prediction of observations with complete records across all explanatory variables, regardless of the existence of pre-observed class labels. Predictions are given as $\hat{Y}_i$, corresponding to the i^{th} observation's prediction, where every label forecast is one of the possible groups contained in G. Intuitively, after the original data space has been demarcated, it then degenerates leaving only the model derived boundaries as marker fields or zones. Each zone corresponds to a specified label, wherein all observations contained within are classified into the boundary specified group. Hence, at the conclusion of the supervised learning process, the training data has been used to calibrate the model which results in estimated boundaries for the various classes.

2.1.2 Classification Models

Statistical procedures and machine learning algorithms that perform supervised learning tasks are aptly called classifiers. As a collective unit, classifiers each perform the same duties, transforming input data into class membership predictions, yet individually, each technique is grounded in theory derived from different assumptions and hypothesis. The work involved in this thesis will revolve around six standard classification models: Classification and Regression Trees, Random Forests, AdaBoost, Naive Bayes, Support Vector Machines, and Neural Networks. To establish a rudimentary understanding of the models and their approaches, a brief introduction to each classifier will be provided. It is the intent of the introduction to familiarize the reader with the theoretical underpinnings of each technique and highlight the diversity in methodology.

Figure 2.1 From left to right, the initial data is plotted and has it's optimal non-linear boundary derived by the classifier. The bounds are then used to differentiate between the two groups and here the accuracy of the bounds can be assessed. Lastly, the bounds themselves can be used to classify any data observation within the data space as either a red or blue class member.

2.1.2.1 Classification and Regression Trees

Decision tree learning is all encompassing phrase that describes rule based partitioning methods. Developed in the early 1980's, classification and regression trees, like most machine learning algorithms, had its usefulness spurned by the advent of high-powered, low cost computing. These rule based techniques rely on the identification of homogenous "splits" derived from subsetting along regions of the input variables (Tibshirani and Freedman, 2009). The dual process of feature selection and split construction are the initial phases in the tree algorithm design. Determining the variable order involves searching among the explanatory variables for fields that will yield the most homogenous splits. A split's quality is assessed through variety of metrics which can include information gain or entropy calculations (Ripley et.al, 2013). Through the use of impurity metrics, each split's level of homogeneity can be quantified. As candidate fields are selected and included in the tree, the rule-based model grows, encompassing a large yet nuanced path along the data space. Because it is a split-based partitioning method, classification and regression trees paths can be easily interpreted. Each path represents a given a set of scenarios that lead to a cluster of similar observations. Following along each path, the model accounts for interactions between variables within the data space. As such, classification trees are an excellent tool for the search of interactions between fields with the inside of an exploratory data analysis framework. Recursive partitioning techniques, do not have any set assumptions that must be followed. That said, the lack of assumptions to be explicitly satisfied does not grant liberty from careful consideration to the application. As a result of the algorithm's construction, variables with a larger number of categories are preferred over variables with fewer levels and when the model is allowed to grow in an unconstrained fashion, the problem of overfitting cannot be avoided. The phenomenon of overfitting occurs when the model grows and complexity and reaches the point where the given data can be perfectly explained by the model yet the complexity mars its ability to make generalized predictions (Izenman, 2008). To protect against the scenarios machine learning practitioners take careful consideration to examine the resulting rules from the tree algorithm. Moreover, to combat the overfitting issue, pruning techniques have been developed that impose penalties for excessive growth beyond

essential branches. Despite its relative simplicity, tree-based methods have been proven to be a reliable and largely attainable technique for model based prediction (Tibshirani and Freedman, 2009).

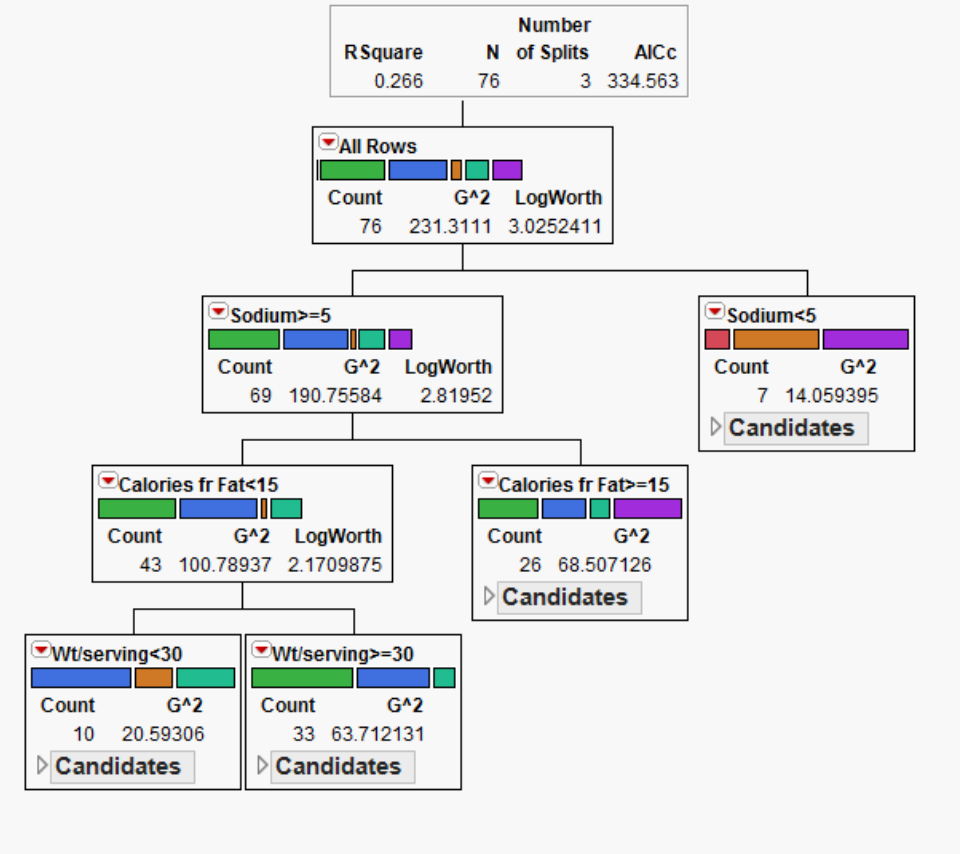

Figure 2.2 A decision tree with rules for differentiating between cereal manufacturers based on a product's sodium content, calories from fat, and weight per serving.

2.1.2.2 Random Forests

Machine learning experts, borrowing from the field of statistics, realized that improvements to tree based predictions could be easily accomplished by incorporating resampling methods, in conjunction with model ensemble techniques, into the modeling framework (Breiman, 2001). This simple modification increased the computational costs and complexity of the modeling procedure, yet has been shown to increase predictive accuracy while maintaining resistance

to overfitting. Dubbed, random forests, the algorithm gained notoriety after Leo Breiman's seminal paper where he described the process of randomly selecting from a fixed training set and allowing only a subset of variables to act as candidates to entry. The algorithm forms a random subspace wherefore features are searched through. As a result of the randomization of both the features and the subset, diversity is induced within the subsequent trees that are created. This process is akin to taking small randomly selected subsections of data along with random subsets of features, fitting a tree model to each subset, and repeating the procedure a predefined number of times. More formally, for some subset of training objects, **N** and features, **X**, at each iteration choose a subset x of size $|x|$ to be the number of input variables to be used in each individual classifier such that $|x| < |X|$. Select $|M|$ to be the number of individual classifiers to be fit by the ensemble. For each classifier, m, we first randomly sample $|N|$ observations from the data, with replacement, and fit a tree classifier without any growth constraints. Each model is given a single vote and the majority voting scheme is employed to determine the class membership of each observation.

Random forests have been shown to be not only efficient on larger datasets, but also outperform other more sophisticated machine learning techniques (Breiman, 2001). Other benefits of random forests include the lack of a need for cross validating the model to develop an unbiased estimate of test set error and the ability of the model to return Gini based variable importance rankings, along with a host of features that are documented in Breiman's original paper. Practitioners have been well served by the variable rankings returned by the random forest procedure. They are a natural result of the tree based construction, are not limited to only categorical or continuous variable types. To accomplish its variable importance ranking, the decrease in Gini for splits under each tree is summed across all trees in the forest and sorted according to the variables which have the largest decrease in Gini. Further advances in the ideas of model aggregation, resampling and computation have spurned even more state of the art techniques that rest on the same fundamental principles as random forests. One such advancement has been the advent of adaptive boosting which will be discussed in the next section.

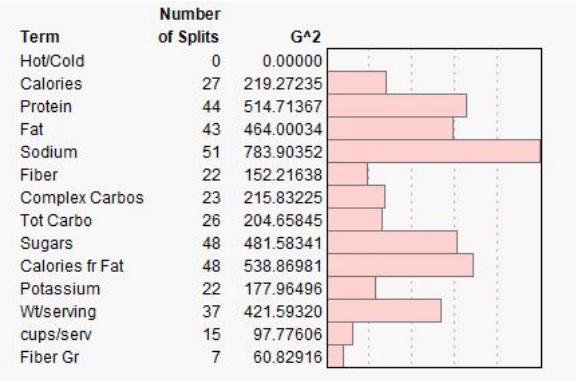

Figure 2.3 Random forest Gini based variable rankings for differentiating between cereal manufacturers.

2.1.2.3 AdaBoost

Boosting is a machine learning approach that supposes the use of many "weaker" models, i.e. low preforming classifiers, crowd sourced to create a single well-informed body of classifiers that will improve predictions by accounting for the collective experience of the group (Freund, 1997). This concept is not unlike the random forest procedure, but is generalized to apply to techniques beyond just classification and regression trees. At the crux, boosting provides a systematic framework for fitting multiple classifiers, reweighting their value according to performance on individual observations. Though the previous statement may imply that the classifiers themselves are being reweighted, within the actual algorithm, the observations are assigned an initial weight, say $w_i = 1/N$. Each observation unsuccessfully classified has its weight reinitialized before the next model fit. Key to the final predictions is the majority vote formula:

$$G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m x) \tag{2.1}$$

The function $G(x)$ serves as an aggregator for the individual predictions of each classifier. Figure 2.4 shows the Adaboost.M1 procedure as described by Tibsherani, Fredman and Hastie

on page 339 of *The Elements of Statistical Learning.*

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Figure 2.4 The Adaboost.M1 algorithm procedure.

Lastly, the authors show that the AdaBoost algorithm reduces to the optimization of an additive model across an exponential loss function where solutions are found through the use of a gradient descent search procedure. As a result of the gradient descent technique and the structure of the problem, it has been shown that random classification noise can have a negative effective on AdaBoost's performance. None withstanding, AdaBoost methods perform well in practice and give comparable results with other ensemble based methods (Tibshirani and Freedman, 2009).

2.1.2.4 Naive Bayes

Probabilistic graphical models are relatively simple techniques that allow for the visual accounting of probability augmented characterizations of networked events. These interconnected events have their stochastic properties modeled through the use of conditionally independent probabilities. This allows for the use of complex computations to be expressed and calculated within the graph theory framework. An even more simplistic version of this technique, specif-

ically applied to supervised learning, is the Naive Bayes classifier. As its name suggests, the naive Bayes classifier is grounded in probability theory with Bayes rule at the crux.

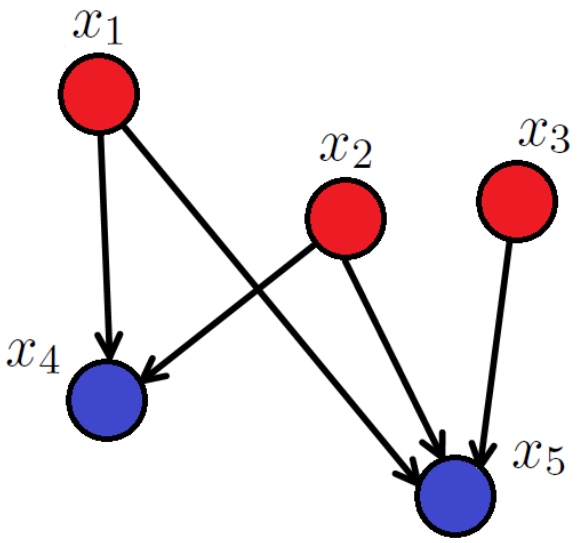

Figure 2.5 A network graph of connected events. The full joint probability can be given by $p(x_1 \cap x_2 \cap x_3 \cap x_4 \cap x_5) = p(x_1) * p(x_2) * p(x_3) * p(x_4|x_1x_2) * p(x_5|x_1x_2x_3)$.

Bayes rule allows for the estimation of conditional probabilities not directly observed using information that can be directly measured and quantified. Under the independence assumption, Naive Bayes classifiers exploit the factorization property the independence structure grants to calculate conditional and joint probabilities of class memberships. Satisfaction of the independence assumption also imposes a 'naivety" assertion that gives equal weight to all features used in the model. This has the unfortunate side effect of making Bayes classifiers susceptible to increased signal noise caused by irrelevant features which may hinder performance (Rish and Watson, 2009). Executing the Naive Bayes classification scheme requires the computation of the maximum a posteriori decision rule which is calculated by selecting the class that yields the largest posterior probability.

$$Class(x_1, \ldots, x_m) = \underset{g}{\operatorname{argmax}}\ p(G = g) \prod_{i=1}^{m} p(X_i = x_i | G = g) \tag{2.2}$$

The above equation quantifies the choice of class as the group membership that maximizes the posterior probability as calculated by looking at the conditional probability of a feature given a specific class. True to its Bayesian nature, the prior information is contained in the $p(G = g)$ term obtained from the observed class memberships in the data. Applying the product across each of these observed conditional probabilities ranks the posteriors with respect to the class of interest so that the one with the highest value can be selected. It has been shown in the past that deviations from the independence assumption makes the numerical estimates unreliable, but do not lead to permutations in the rankings of posterior probabilities and therefore the final output, an estimated class membership, are often reliable predictions. Due to its simplicity and efficiency, the Naive Bayes classifier even has an Apache Mahout big data implementation that works for gigabyte and terabyte sized datasets (Rish and Watson, 2009).

2.1.2.5 Support Vector Machines

Hermann Minikowski, a German mathematician, is responsible for creating the "separating hyperplane theorem". This theorem purports that if given two disjoint, closed, convex sets, say A and B, with properties such that one set is compact, then these two sets have a pair of points p and q where one point lies in each set such that a hyperplane exists perpendicular to the line segment between points p and q. Minikowski's assertion shows that under certain conditions there will exist an N-dimensional line segment that will separate convex sets. In machine learning, the exact conditions do not hold for every set, but the concept of searching for a separating hyperplane motivated the creation of a technique called support vector machines. This technique involves finding a solution to a quadratic programming problem of the form:

$$\min_{w,b} \tfrac{1}{2}\|\mathbf{w}\|^2$$

s.t.

$$y_i(w \cdot x - b) \geq 1$$

Where w is a normalizing vector, b is a constant, x is the value of the observation, and y is the class of the observation. This quadratic program can be relaxed with Lagrangian multipliers to make the problem more tractable. In practice, most boundaries between classes are not separable due to overlap, which hinders the search for a support bound that maximizes the distance between the support vectors. This can be compounded with the addition of non-linear boundaries. To account for this scenario, what is known as a "kernel trick" can be applied to the data by recasting the datum into a higher dimension and searching for a linear bound. When the data is returned to its original dimension, the resulting higher dimensional linear bound is now non-linear. This powerful and clever mathematical transformation has proven to be extremely useful in practice (Tibshirani and Freedman, 2009). Support vector machines have been shown to preform very well in practice for both binary and multi-class prediction problems.

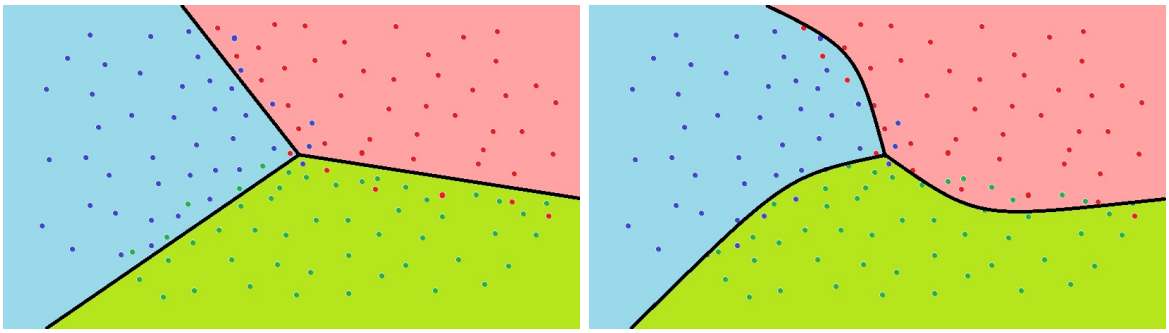

Figure 2.6 Sample linear and non-linear bound for a support vector machine.

2.1.2.6 Neural Networks

Originating to emulate the biochemistry of the brain, neural networks is one of the most popular machine learning algorithms in use today. As a model, it attempts to focus on linear combinations of the inputs and then characterize the response variable as a function of these linear combinations. At the advent of its creation, neural networks became very popular,

yet as it was studied in more detail some of its short-comings became more apparent. Neural networks have an inability to handle mixed data well, have no integrated procedure for handling missingness, can be biased easily due to outliers, do not scale well, aren't interpretable and do not handle noisy input variables with any degree of intelligence (Arel, 2010). Beyond those shortcomings, neural networks also have a tendency to overfit to the existing training data (Kotsiantis, 2007). As a result, the ubiquitous use of neural networks waned toward the end of the 1990's due to issues with its performance. Recently, big data repositories and extensions of neural networks aptly named "deep learning" has brought on a resurgence of the technique's popularity. When given a sufficient amount of data, neural networks ability to model nuances allows for the quick search through large feature spaces for patterns that traditionally not be distinguishable. Within small data contexts, these same patterns often serve as noise, yet with sufficient data these patterns give marginal improvements in accuracy which can be substantial in the aggregate. Deep learning algorithms attempt to improve predictions by layering neural network models into a machine that conceptualizes a hierarchy of features in the data (Arel, 2010).

2.2 Model Assessment Metrics

2.2.1 Model Validation

Model evaluation is crucial to the machine learning, data mining process because it is the method by which the legitimacy of models are tested. As such an important phase in the learning process, the evaluation must be simultaneously both objective and robust. To fulfill the objectivity necessity, quantitative measures, be they information-theoretic based or matrix reduction, are used to provide a singular numerical representation of the quality of a model. Representing this model quality with error or misclassification rates is standard, just as well as citing their inverse, accuracy. Scenarios with imbalance will be discussed later; however, performing model evaluation for multi-class problems is not just a straightforward extension of the binary case. Many techniques do not have multi-class extensions, particularly ones set in an information theoretic frame. Therefore any measure utilized in a multi-class situation

must be robust beyond the binary case. The following sections will introduce commonly used metrics and discuss their use in practice.

2.2.2 Contingency Tables

The consortium of performance measures defined on 2x2 confusion matrices and their more general kxk counter parts can be parsed based on how the off diagonal misclassification knowledge is processed (Wei et.al, 2010). Measures derived from information theory treat the actual class as a model input and their corresponding predictions as output. The classifier, acting as a communication channel between input and output, allows for the use of information theoretic tools which seek to characterize the amount of entropy or information loss in a given confusion matrix (Moreno and Albacete, 2010). In essence, the confusion matrix is acting as a random variable and its information content measured accordingly. These measures usually afford a high degree of matrix discrimination, which serves well to detect differences in misclassification rates from similar matrices, an asset when class distributions are skewed in favor of one class. Unfortunately the nature of information theoretic derivations, which rely heavily on non-trivial differential entropy, make extensions of these measures difficult to construct as supported anecdotally by their scarcity in the multi-class prediction assessment literature. The other branches of measures rely on confusion matrix reduction and transformation to glean misclassification information (Moreno and Albacete, 2010). Individual elements and sums are manipulated to reduce the k times k matrix entries into a single number that represents the classification accuracy. This simplicity often comes with a cost, as information loss is inevitable when reducing a kxk table into a single number (Chauvin et.al, 2000).

We will briefly discuss the some of the more common measures applied to two-class and k-class prediction. Let C^k denote a confusion matrix or the contingency table of actual class labels by their model predictions, with c_{ij} representing the number of cases with true label i classified into group j. A sample construction of a 2x2 confusion matrix is given in Table 2.1.

		Predicted		
		Class 1	Class 2	Total
Actual	Class 1	c_{11}	c_{12}	$c_{11} + c_{12}$
	Class 2	c_{21}	c_{22}	$c_{21} + c_{22}$
	Total	$c_{11} + c_{21}$	$c_{12} + c_{22}$	N

Table 2.1 A 2x2 Confusion Matrix denoted as C^2.

2.2.3 Two-Class Evaluation Measures

Accuracy As the current de-facto accuracy measure, overall Accuracy is simple to calculate and interpret. Within a contingency table, successfully classified observations appear along the diagonal. Accuracy, therefore, is simply the proportion of correctly classified observations divided by the total number. Following the notation in Table 2.1, Accuracy is defined as:

$$Accuracy = \frac{c_{11} + c_{22}}{c_{11} + c_{12} + c_{21} + c_{22}} \tag{2.3}$$

Recall, Precision, and the F-measure Despite being easily attainable, this 3-tuple of measures is less commonly used. As per Table 2.1, classifier Recall is calculated from the number of correct Class 1 matches divided by the total number of actual Class 1 cases. Similarly, Precision aptly describes how precise a model is by dividing the number of correct Class 1 matches by the total number of predicted Class 1 instances. The F-measure supplements them by reducing both measures into a single number by producing the harmonic mean between Precision and Recall. The usefulness of these measures has largely been restricted to document retrieval and similar applications. Their formula is as follows:

$$Precision = \frac{c_{11}}{c_{11} + c_{21}} \tag{2.4a}$$

$$Recall = \frac{c_{11}}{c_{11} + c_{12}} \tag{2.4b}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{2.4c}$$

Sensitivity and Specificity Reporting the sensitivity and specificity of a laboratory diagnostic test is a generally accepted practice in the medical literature because of their direct

relation to type I and type II error. Sensitivity is identical to the Recall measure discussed erstwhile and is calculated for Class 1. Specificity judges a classifier's ability to correctly identify Class 2 instances. To avoid misleading conclusions, both numbers are reported when assessing testing procedures. At this time, aside from averaging across each class, no well-established multi-class generalization exists. Explicitly stated the formulas for Sensitivity and Specificity are:

$$Sensitivity = \frac{c_{11}}{c_{11} + c_{12}} \tag{2.5a}$$

$$Specificity = \frac{c_{22}}{c_{21} + c_{22}} \tag{2.5b}$$

ROC and AUC Receiver Operator Characteristic (ROC) curves and the Area Under the Curve (AUC) are common measures in medicine, machine learning, and a host of other fields (Arun and Sheshadri, 2012) that want to take advantage of the well behaved statistical properties and leverage the graphical nature of the technique. Unlike the previously mentioned techniques, ROC curves are not defined on a single confusion matrix but on the class probability estimates. In combination with the probability estimates, if given a set probability threshold value, a model's sensitivity can be plotted on the y-axis against the false positive rate creating the curve. Naturally, the area under the curve could serve as a measure of model quality, since at perfect accuracy both ROC axis measures are maximized suggesting that larger areas are superior. This single number reduction has prompted hopeful researchers to seek meaningful multi-class extensions of the AUC measure. One such extension, the Volume under the Surface, has been derived but was shown to be particularly unwieldy (Moreno and Albacete, 2010). It wasn't until Hand's work in 2009 did the entire foundation of using AUC as a measure become suspect. Hand boldly states that "...using the AUC is equivalent to using different metrics to evaluate different classification rules." Recently other authors have cited his work and published extensions or proposed their own alternative solutions for AUC's incoherency.

2.2.4 k-Class Evaluation Measures

Matthew's Correlation Coefficient First introduced in 1975 by Brian W. Matthews for the 2x2 case, this measure has been carefully studied and shown to have connections to the χ^2 distribution (Chauvin et.al, 2000). The measure has some other notable characteristics, such as an intuitive [-1,1] range where the bounds represent perfect misclassification and perfect classification, respectively. MCC calculates a value of 0 for confusion matrices that indicate the classifier preformed the classification randomly. Findings have discussed MCC's relationship with Confusion Entropy, a measure discussed later, have been explored for fruitful results (Jurman and Furlanello, 2010). Though MCC has been gaining more traction as one of the best binary classification task measures, how it performs in multi-class settings with unbalanced groups has not yet been well studied. The formal expression is as follows:

$$MCC = \frac{\sum_{i,l,m=1}^{k} c_{ii}c_{ml} - c_{li}c_{im}}{\sqrt{\sum_{k=1}^{n}(\sum_{k=1}^{n} c_{lk})(\sum_{\substack{f,g=1 \\ f\neq k}}^{k} c_{gf})}\sqrt{\sum_{i=1}^{k}(\sum_{i=1}^{k} c_{il})(\sum_{\substack{f,g=1 \\ f\neq k}}^{k} c_{fg})}} \tag{2.6}$$

Relative Classifier Information RCI is an information theoretic approach designed expressly to summarize how distinctly classes have been demarcated (Wei et.al, 2010). This measure has a deceptively intuitive range of [0,1] where large values indicate better classification performance; however, the measure's construction does not account for actual accuracy

only the uniformity of the predicted classes. Stated explicitly, the formula for RCI is given as:

$$RCI = \frac{H_c}{H_d} \tag{2.7a}$$

$$H_d = -\sum_{i=1}^{n} \frac{\sum_{i=1}^{n} c_{il}}{n} log(\frac{\sum_{i=1}^{n} c_{il}}{n}) \tag{2.7b}$$

$$H_o = \sum_{j=1}^{n} \frac{\sum_{k=1}^{n} c_{kj}}{n} H_{oj} \tag{2.7c}$$

$$H_{oj} = -\sum_{i=1}^{n} \frac{c_{ij}}{\sum_{k=1}^{n} c_{kj}} log(\frac{c_{ij}}{\sum_{k=1}^{n} c_{kj}}) \tag{2.7d}$$

$$H_c = H_d - H_o \tag{2.7e}$$

Confusion Entropy Continuing within the information theory framework, Wei et.al. define their measure, Confusion Entropy, on multi-class confusion matrices by focusing on all available information contained in the off diagonal entries. As a result of their careful derivation, they created a measure that discriminates among matrices better than any previous measure to date (Jurman and Furlanello, 2010). The resolution of Confusion Entropy's separations is so pronounced that the measure values can't assign a unique value to all cases that represent random classification like its MCC counterpart. For this entropy measure, small values represent less information loss and better classification, and in practice this fact must be kept in the forefront because of its counterintuitive nature. The Confusion Entropy is defined as:

$$CEN = \sum_{j=1}^{n} P_j \sum_{\substack{k=1 \\ k \neq j}} h_{2(n-1)}(P_{jk}^{j}) + h_{2(n-1)}(P_{kj}^{j}) \tag{2.8a}$$

$$h_b = P(x)log_b(P(x)) \tag{2.8b}$$

$$P_{ij}^{j} = \frac{c_{ij}}{\sum_{k=1}^{n} c_{jk} + c_{kj}} \tag{2.8c}$$

$$P_{ij}^{i} = \frac{c_{ij}}{\sum_{k=1}^{n} c_{ik} + c_{ki}} \tag{2.8d}$$

$$P_{jij} = \frac{\sum_{k=1}^{n} c_{jk} + c_{kj}}{2\sum_{k,l=1}^{n} c_{kl}} \tag{2.8e}$$

$$P_{ii}^{i} = 0 \tag{2.8f}$$

Balanced Accuracy Balanced Accuracy is the Recall for each class, averaged over the number of classes. As an assessment tool it is intuitively simple, the predictive quality is measured for each class independently and aggregated. Balance accuracy derives all of its information from the diagonal elements and the row sums.

$$Balanced\ Accuracy = \frac{\frac{c_{11}}{c_{11}+c_{12}} + \frac{c_{22}}{c_{21}+c_{22}}}{2} \tag{2.9}$$

(2.10)

G-Mean Similar to Balanced Accuracy, the Geometric Mean focuses only on the Recall of each class. What differentiates this measure from balance accuracy comes from the way the class recall is aggregated; multiplicatively instead of additively across each class.

$$G - Mean = (\Pi_{i=1} r_i)^{\frac{1}{k}} \tag{2.11a}$$

$$r_i = Recall\ for\ Group\ i \tag{2.11b}$$

The multi-class measures show much more promise than their 2x2 counterparts with regards to practical application in the presence of imbalance, yet the field is still open for measures that can provide simplicity, are mathematically coherent and extendable beyond two classes.

2.3 Background and Formalization of the Class Imbalance Problem

In this section the class imbalance problem will be formalized and followed with a discussion of its effects on supervised learning tasks.

2.3.1 Formalization and Definitions

Suppose for a given dataset, $\{\mathbf{X}, \mathbf{Y}\}^n$, we recall that the Y^n component is a **n**-dimensional collection of singleton elements from the set **G**, whose units are distinct labels g_1,...,g_k. Here we introduce the set **P**, a container for the proportions of each class distribution. In similar fashion to the erstwhile defined sets, the elements of **P** are denoted as p_1,p_2,p_j,...p_k. Each p_k proportion has a value that ranges from 0 to 1.

Classical machine learning algorithms assume the response variable has an equal number of observations within each class. The multi-class imbalance problem can be stated simply as any deviation from this assumption where at least one class proportion p_i is not equal to the other proportions when $k > 2$. It is obvious that in practice, most if not all tasks will exhibit imbalance due to the inability to control the outcome of a model, experiment or procedure. It is true however, that since data mining tasks involve the development of the model ex post some procedure, and it is possible to select an equal number of observations of each class as long as the researcher is comfortable ignoring observations. Imbalance can also occur in instances where limitations are due to collection of data due to cost or privacy. Returning to our previous thought, class imbalance, according to its strict definition, occurs frequently in practice, and yet does not have much effect on the outcome unless the imbalance reaches some threshold. Unfortunately, it is at this junction where the objectivism of defining class imbalance departs. Because imbalance can occur at varying degrees, there is no set standard wherefore we can definitively say that the imbalance within a class variable is indeed a problem. Currently, at best we can hope for the development of a threshold value that will indicate if a response subset suffers from class imbalance to such a degree that it will have some impact on the modeling process (Japkowicz, 2000). At this point in time there is no such indicator.

None withstanding, despite there not existing a well-established cut-off, previous publica-

tions have established definitions for two common imbalance scenarios (Wang and Yao, 2012). For any multi-class problem, a "multi-minority" case is one where a single class has a significantly larger proportion than the average size of all other classes. The converse situation is where a single class has a significantly smaller proportion than the average size of all classes. This is deemed the "multi-majority" case. This situation can be formalized as $p_{min} << \overline{p}$ where $\overline{p}$ is the average proportion across all classes. Likewise for the multi-minority case $p_{maj} >> \overline{p}$ where $\overline{p}$ (Wang and Yao, 2012).

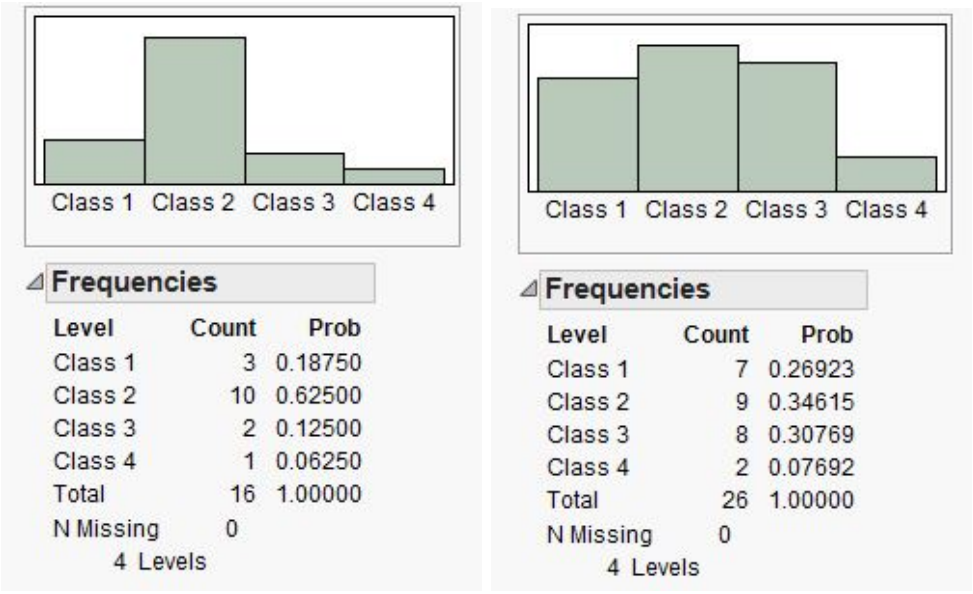

Figure 2.7 Multiple minority and multiple majority imbalance scenarios.

Wang and contributing authors point out that both forms of imbalance negatively affect both overall and per class performance. They found that multi-majority cases to be the more harmful of the two, hindering common data based imbalance solutions, ultimately leading to overfitting issues. In addition, when multiple minority classes exist, random undersampling greatly reduced the majority class performance. In the next section we will discuss the effects of class imbalance in more detail.

2.3.2 Effects of Class Imbalance

Class imbalance influences data mining tasks by proxy. In the presence of imbalance, algorithms can be initialized, their procedure will run, and converge can be met; therefore, the effect of imbalance are largely symptomatic. Despite a non-terminal prognosis for models trained under imbalanced distributions, unequal class distributions exacerbate already troublesome data mining issues such as over-shadowing minority classes when there exists concept complexity, introducing additional training bias when building models with a small sample sizes, and invalidating commonly used accuracy measures (Japkowicz, 2000).

2.3.2.1 Concept Complexity

When majority and minority classes exhibit a low amount of separability within the feature space, the data is said to express a high degree of concept complexity. The idea of separability communicates the degree in which observations share similar values along fields in the feature space. With each similar value, learning techniques must search an alternative variable or linear combination of fields to separate the observations. In the presence of imbalance, this overlap creates blurred boundaries between the classes (Drummond and Holte, 2005). When combined with an accuracy driven algorithm, it creates a situation where the minority class observations can be ignored (Drummond and Holte, 2005; Wang and Yao, 2012; Dongre and Malik, 2013). This will be a critical theme within this body of work.

As a generalized term, concept complexity also describes the linearity of the class boundaries. Linearly separable boundaries are the holy grail of modeling bounds. Nearly all algorithms, either search based or statistically grounded, can find an optimal linearly separable plane where groups can be partitioned (Tibshirani and Freedman, 2009). These bound also happen to be easily interpretable with explanations that are conducive to creating classification rules. Unfortunately, when bounds are non-linear, the "classification bounds" for groups can take any shape or form, which increases the computational effort required to carve boundaries to some sensible approximation. A further complication can occur when these clusters of minority observations do not reside in one centralize location within the data space. Scattered

pockets of disjoint minority classes can exist with non-linear bounds and in situations where the overlap of majority classes seep into the minority class segments algorithms err towards ignoring the minority class segments (Wang and Yao, 2012).

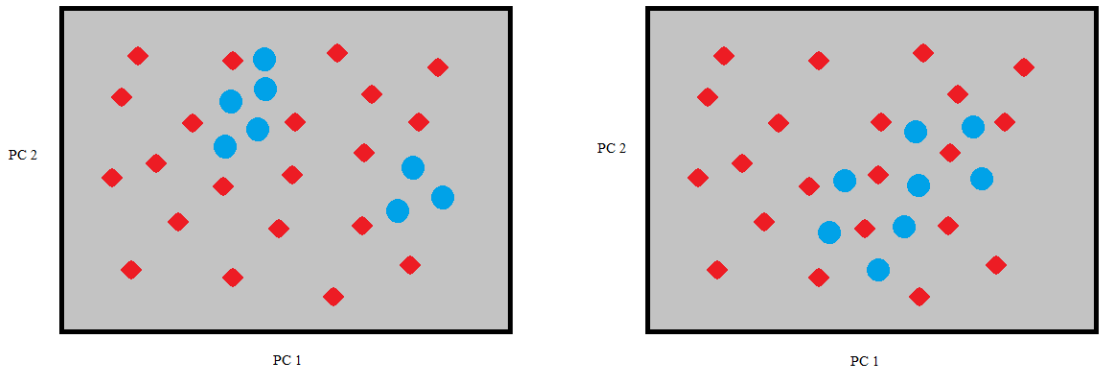

Figure 2.8 Both figures are suffering from concept complexity. On the left is a dataset with small disjoints, while the figure on the right suffers from significant class overlap.

2.3.2.2 Small and Noisy Data

Noisy data can be described as data that have been gathered in some ill-prepared manner, incorrectly labeled, or contain features that provide spurious information. Minority class observations, which already exhibit sparse representation, are especially prone to noise by biasing the boundaries away from their true limits. As a consequence of noise, classes that originally may be linearly separable now increase in concept complexity and bring along with it all the subsequent ramifications (Garcia et.al, 2007).

It is intuitive that the fewer data points in the datum, the easier it is to separate groups, the faster algorithms will converge and higher the likelihood for the boundaries to be linearly separable; however, as a consequence of having a reduced sample sizes, group demarcations found will not be generalizable. With imbalance extant, the metes created to differentiate the

minority class observations may not approximate the true boundaries because of sampling bias.

2.3.2.3 Evaluation

As discussed previously, model evaluation is an integral part of the data mining process. When comparing and contrasting the model predicted class memberships with the actual class groupings any measure utilized should consider both the overall accuracy of predictions along with the individual class recalls. In the presence of imbalance, accuracy measures that focus on overall performance will have a tendency to ignore minority classes because as a group they do not contribute much to the general performance (Zeno et.al, 2011). As a classic example, given ninety-eight observations with "positive" labels, a single "negative" observation, and a single "neutral" labeled instance, if the latter two points are not conspicuously separated in the data space then most classifiers would be well suited to create a rule that classifies all observations as a positive group member. The learning rule would achieve ninety-eight percent accuracy, but effectively provide no new knowledge if the initial objective was to gain insight and demarcate boundary lines between the three classes. While the value added of this classification model would be nil, our evaluation criteria returns a value that suggests directly the opposite. In effect, when information about each class is integral, class imbalance severely hinders the effectiveness of traditional accuracy as a performance measure.

Unfortunately, it is still common practice for scholars to report measures that account only for the overall performance of the classifier (Galar et.al, 2012). This is a result of a lack of consensus for the choice of measures in the presence of skewed class distributions. Many measures lack the ability to be generalized to the multi-class case, which hinders their use beyond binary classification and therefore are invalid for multi-class imbalance problems. As a further consequence of complexity, implementations of non-matrix reduction techniques are uncommon and restricted to a few programming languages. Therefore there is a gap in the literature for any measure that can account for accuracy across all classes, is robust to the cardinality of the class set, and possesses a form that is easily implementable.

2.4 Current Approaches for Class Imbalance Prediction

Previous investigations into imbalance have shown that the effects of skewed class distributions are a function of the degree of imbalance, the amount of overlap between the minority and majority classes, the overall size of the data and the classifier itself (Wang and Yao, 2012; Japkowicz, 2000). Methods to address class imbalance attempt to do so by mitigating the influence of one of those four characteristics of the modeling process. These characteristics form the basis for the two general approaches that involve either data space manipulation, algorithm modifications or an amalgamation of both. This section we will discuss common approaches to the class imbalance problem in more detail.

2.4.1 Data Methods

A straight-forward procedure involves rebalancing the class distributions through resampling the data space. These methods involve either oversampling the minority class or undersampling the majority class. In their simplest form, oversampling and under sampling involve the random selection of data observations already extant in the data. This has the consequence of being computationally quick, however both can potentially and often do bias the datum in unintentional ways. Oversampling the minority class has been known to increase the chances of overfitting because observations within the minority class are exact replicas of one another. Random undersampling does not have this effect, but can potentially discard useful observations. To account for the shortcomings, techniques such as synthetic minority oversampling technique and selective preprocessing of imbalance data were introduced. SMOTE is a k-nearest neighbor approach to minority class oversampling. The hope is that the overfitting problem can be sidestepped by generating new instances from a random interpolation of existing minority members. SPIDER is a hybrid technique that involves both over sampling the minority class and under sampling the majority class through an intelligent two-phase process of identification and preprocessing. The first phase begins with the identification of misclassified instances using k-nearest neighbors. The second phase then decides whether to amplify minority class instances, amplify the minority class instances and relabel majority class instances, or just re-

label majority class instances. As a whole, there has not been an extensive survey of the effects of data pre-processing on imbalanced data prediction. As classifier independent techniques, resampling methods can be applied directly to the data set and used in conjunction with any classifier technique which is a boon, yet it is an unfortunate circumstance that there does not exists a single repository containing open source robust implementations of these techniques.

2.4.2 Algorithm Methods

Algorithm approaches to the class imbalance problem make use of ensemble techniques and clever cost assignments for class observations. The latter, cost sensitive learning is a procedure that reweights observations according to the relative cost of misclassification. Within the context of class imbalance, minority class observations are given substantially higher misclassification costs than their majority class counterparts (Galar et.al, 2012). It is this reallocation of misclassification errors towards the minority class the forces algorithms to account for them with some form of equality. Some algorithms benefit by the direct incorporation of the cost structure into their designs. In other instances, costs are incorporated ex post to determine which modeling procedure performed the best with respect to minimizing the cost of misclassification. A major drawback to cost sensitive techniques become apparent through their need to have misclassification costs clearly defined when in practice an objective and quantifiable cost structure may not exist (Galar et.al, 2012). For example, in medical applications when trying to make predictions across several different terminal illnesses, if we assume that the quality of life is constant across each, the cost of misclassifying the patient into a rare terminal disease as opposed to a common terminal illness is not directly identifiable.

The former of the two approaches involves the exploitation of model diversity to develop a crowd sourced prediction of class memberships. Multiple classifiers are trained from the data and combined using some standardized voting scheme to determine the final class estimate. Ensemble techniques are often not used as standalone techniques for dealing with imbalance problems as they were initially developed predominantly as prediction improvement routines, however scholars have been able imbed misclassification costs into the ensemble framework for positive gains in accuracy across the minority class which has spurred their use in the class

imbalance literature (Galar et.al, 2012; Wang and Yao, 2012).

2.5 Data and Computing

The UCI Machine Learning Repository at the University of California at Irvine is home to a collection of data sets used for the evaluation and testing of machine learning algorithms. These data sets span across many academic fields such as engineering, epidemiology, business, and biology. The datum in this repository exhibits many of the features, or better described as shortcomings that data in the real world harbor. Centralized one place, the repository's data sets can suffer from data structure issues such as low statistical variation within fields and improper formatting to more technical maladies such as high dimensional noise, missingness, class imbalance, which can all be detrimental to the performance of algorithms trained on this data. Because of the variety and diversity of the variance-covariance structures within these data, the utilization of these data sets for machine learning algorithm validation offers a robust picture into a technique's performance, setting the UCI machine learning data repository as the standard for which machine learning algorithms are vetted. Particular for this research, the data sets chosen were specifically selected for their diversity with respect to class imbalance. The data sets of interest were not only binary, but multi-class in nature and possessed various forms of non-uniformity within the response variable's class memberships. This allowed our simulation studies to present results across a wide array of real-world scenarios. To augment the data diversity further, a supplemental dataset, "diamonds" was added from the statistical visualization literature. The diamonds data was compiled from http://www.diamondse.info/ in 2008 by then graduate student Hadley Wickham and contains both quantitative and categorical variables. By careful consideration, the datasets used in this research and the results derived from them should be extendable onto other modeling scenarios with similar data structures.

All statistical computations for this work utilize open-source software freely available in the public domain. The primary programming languages used to produce this research were R and Java. The R 3.0 64-bit software environment acted as the primary work horse for simulation, algorithm, and measure implementation. In conjunction, the RStudio integrated development environment capabilities were leveraged for its improved graphical interface, storage, and

Sweave integration. Each machine learning algorithm used were called from their respective pre-packaged implementations as downloaded from CRAN, the comprehensive R archive network, which serves as a repository for publicly released functional implementations of statistical procedures. The packages which contain the models explored in this research are cited in the reference section. Lastly, to perform model based instance selection, Java implementations of class balance accuracy and both the greedy addition and subtraction instance selection techniques were created. Calculations and code compiling was shared across a variety of computers, however the predominant analysis machine was a Windows 7-based computer with a i7-2600 quad-core processor with 16 GB of dedicated memory.

Name	# of Classes	Total # of Obs.	Number of Obs. in Class
Anneal	5	798	608 \ 88 \ 60 \ 34 \ 8
Audio	24	226	57 \ 48 \ 22 \ 22 \ 20 \ 9 \ 8 \ 6 \ 4 \ 4 \ 4 \ 3
			2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1
Balance	3	625	288 \ 288 \ 49
Ecoli	8	336	143 \ 77 \ 52 \ 35 \ 20 \ 5 \ 2 \ 2
Flare	6	1389	396 \ 327 \ 287 \ 212 \ 116 \ 51
Glass	6	214	76 \ 70 \ 29 \ 17 \ 13 \ 9
Hepititis	2	112	93 \ 19
Nursery	5	12960	4320 \ 4266 \ 4044 \ 328 \ 2
Opti	10	5620	572 \ 571 \ 568 \ 566 \ 562 \ 558 \ 558 \ 557 \ 554 \ 554
Page	5	5473	4913 \ 329 \ 115 \ 88 \ 28
Pendigits	10	10992	1144 \ 1144 \ 1143 \ 1143 \ 1142 \ 1056 \ 1055 \ 1055 \ 1055 \ 1055
Sat	6	6435	1533 \ 1508 \ 1358 \ 707 \ 703 \ 626
Segment	7	2310	330 \ 330 \ 330 \ 330 \ 330 \ 330 \ 330
Soy	19	266	40 \ 40 \ 40 \ 20 \ 20 \ 16 \ 10 \ 10
			10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 0 \ 0 \ 0 \ 0
Yeasts	10	1484	463 \ 429 \ 244 \ 163 \ 51 \ 44 \ 35 \ 30 \ 20 \ 5
Diamonds	5	53940	21551 \ 13791 \ 12082 \ 4906 \ 1610

Table 2.2 Data set descriptions for the 16 data samples used in this research.

CHAPTER 3. A GLIMMER OF HOPE FOR MULTI-CLASS ACCURACY MEASUREMENT IN THE PRESENCE OF CLASS IMBALANCE

Our discussions in Chapter 3 will include the motivation of this research work through a guided tour of the shortcomings of current multi-class model evaluation metrics. Afterward, we will formally introduce Class Balance Accuracy as an alternative performance indicator for measuring classification error in the presence of class imbalance and vet its usefulness with simulation results.

3.1 Introduction

Assessing classifier performance from a broad, overall perspective has traditionally served data mining practitioners well, yet as the applications of data mining have become more ubiquitous, machine learning algorithms have begun to be applied to scenarios that challenge their fundamental assumptions. One such assumption requires that there be an equal number of observations from each group in the target variable (Japkowicz, 2000). With respect to model evaluation, a failure to satisfy this assumption prevents many commonly used metrics from providing meaningful insights into a model's performance. When performing classification, there is often a dual goal to be accomplished where we seek the successful partitioning of the data space into pooled boundaries that differentiate classes and do so in a manner that minimizes misclassification error (Galar et.al, 2012). When learning under imbalance, the tendency of machine learning algorithms is to divide the data space in a way that maximizes the overall classification rate irrespective of the intent to discriminate between groups. This can create a scenario where a model with high overall accuracy may have very little contradistinctive power,

and when the degree of imbalance is severe enough, there may be no value provided by the model.

Accuracy measurements, when applied to multiclass prediction results, can be faced with situations where they are unable to differentiate between multiple models. This is a consequence of when measure formulas neglect off-diagonal information and only account for the on-diagonal cells within a contingency matrix (Zeno et.al, 2011). The following figure is an example of one such case. In the first plot, we have three groups graphed according to their X and Y values. Within the range of 0 through 25 on the x-axis, we have 500 data points of both blue and red hue. Between x-values 25 and 50, there are 500 red group observations and 100 blue group points. From 50 to 75 on the x -axis, there are 100 green observations and 500 red. Lastly, between the x-values of 75 to 100 we have an equal number of red and green observations, both with 500 data points. In the second and third plots, we have two alternative models: one that predicts every observation into the red group and another that creates a separate partition for each group. By construction, both models have the same overall accuracy of 62.5%, which is derived from taking the 2000 correctly classified observations and dividing them by the total number of observations, 3200; however, in only one of the two models can the classes be discerned from the classifier boundaries. Though this example may seem extreme, machine learning algorithms do indeed seek bounds that maximize overall accuracy therefore the likelihood of attaining an all red straw model is not beyond reason. It should also be noted, that if only one observation from either of the minority classes were to be relabeled as a red observation, the all red model would have superior overall accuracy performance though it still would not provide any new or useful information for differentiating between classes.

Alternative model performance indicators have been proposed for use in the presence of class imbalance, yet many have undesirable properties which hinder their widespread use. As mentioned previously, the formula for accuracy focuses solely on the diagonal entries omitting relevant off diagonal information (Zeno et.al, 2011). Hence, accuracy should only be used in situations where overall performance is important and the class distributions are uniform. An intuitive alternative would be to average the accuracy of each class which has a formalized name, Balanced Accuracy. Though the logic is sound, since the measure focuses only on

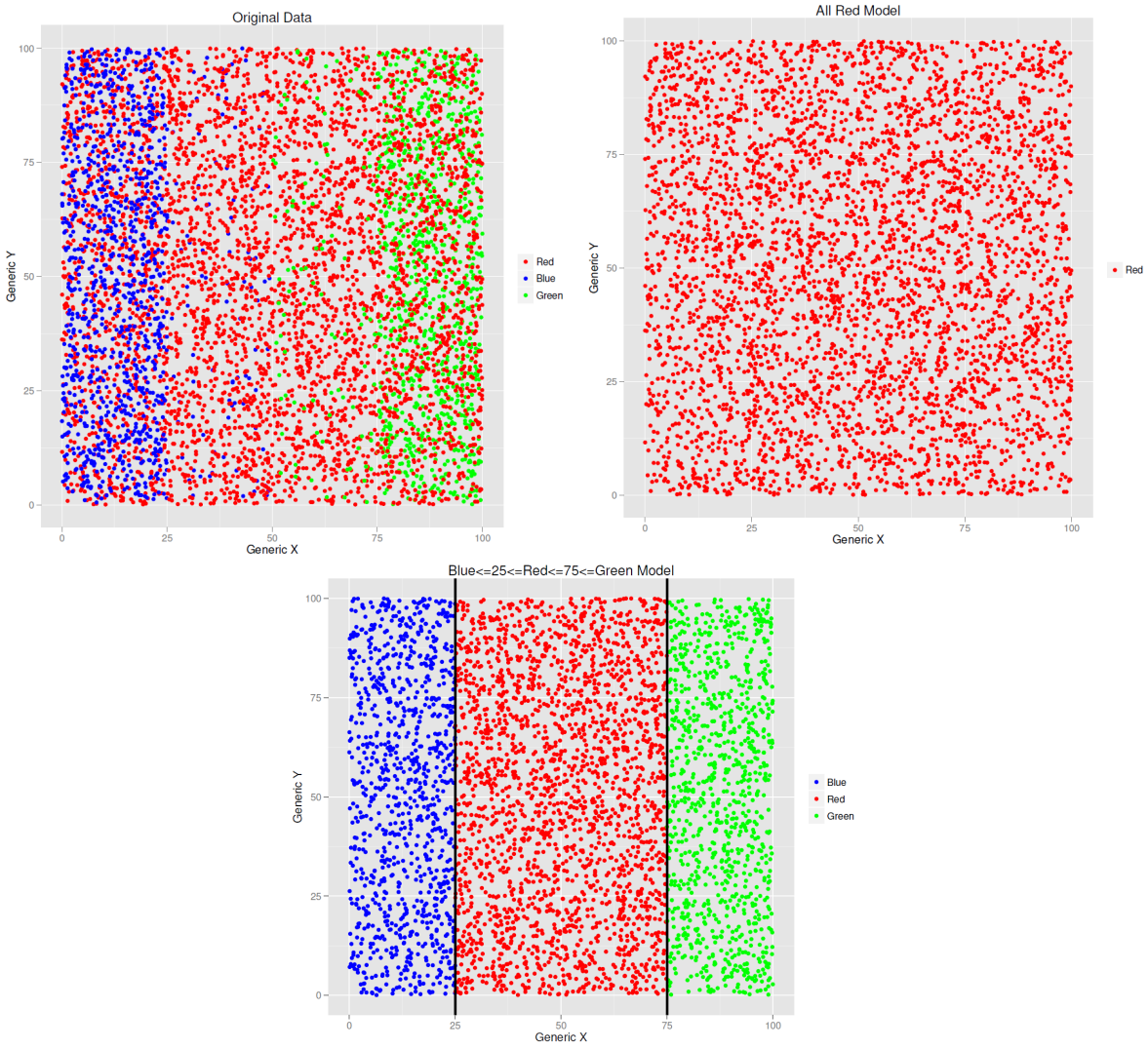

Figure 3.1 A data visualization of all red and class partitioned models derived from the original data set on the top, left. Both models have the same level of accuracy, 62.5%, but clearly divide the data space differently. The Class Balance Accuracy for the all red and class partitioned models are 20.8% and 50% respectfully.

recall it will neglect how well the classifier is actually performing the predictions, i.e. its precision. A loss in the measure's discriminatory power is a direct result of this oversight, and manifests itself when trying to compare two models with similar per class performance. The use of Balanced Accuracy is not wide spread in the class imbalance literature, likely because of the aforementioned shortcoming. The Geometric mean or G-Mean has received some use in literature, but due to its multiplicative nature, algorithms that completely misidentify one class will receive a G-Mean assessment value of zero. In multi-class, imbalanced learning tasks, this level of hypersensitivity is too restrictive. Relative Classifier Information, a measure discussed previously is inadmissible because of a hazardous quality were both perfect misclassification and perfect classification return the same value. Other more traditional measures, such as Sensitivity and Specificity are called "class dependent" and their use has been frowned upon by the imbalanced data community (Weng and Pool, 2006). Furthermore, despite recent attempts to extent AUC to k-class domain, the recent incoherency issues raised require any decisions based on this measure to rightly be subject to additional scrutiny (Hand, 2009; Moreno and Albacete, 2010; Yuan et.al, 2010).

At its crux, the search for an admissible k-class evaluation metric for imbalance tasks revolves around finding a measure that is class independent, scalable to any number of classes, incorporates off diagonal information, balances minority class sensitivity, all while maintaining relative simplicity. It is here that we propose Class Balance Accuracy as a performance measure suitable for use in the presence of multi-class imbalance.

3.2 Definitions, Properties and Proofs

We begin first with a generalization of the 2x2 confusion table. Again, allow C^k to denote a kxk confusion matrix or contingency table of actual class labels aligned by their model predictions, with c_{ij} representing the number of cases with true label i classified into group j. A valid confusion matrix will constrain the model and the output classes to the same set, therefore $i, j \in G$, where G denotes the set of all possible class labels. The cardinality of G, $|G|$, will be the total number of classes, k. Ergo, $i, j = 1, 2, ..., k$. For multi-class classification the number of classes, k, must be greater than or equal to 3. So under this construction,

the confusion matrix is guaranteed to be a square matrix with an equal number of rows and columns. Row and columns sums of a given index are attained by adding across all groups of the remaining index. Therefore the number of actual cases in group i will take the general form:

$$c_{i\cdot} = \sum_{j=1}^{k} c_{ij} \tag{3.1}$$

Likewise, column sums will represent the total number of data observations predicted as class j and have the form:

$$c_{\cdot j} = \sum_{i=1}^{k} c_{ij} \tag{3.2}$$

The grand total of data observations, N, will be the summation of all matrix entries as given by

$$N = c_{\cdot\cdot} = \sum_{i=1}^{k} \sum_{j=1}^{k} c_{ij} \tag{3.3}$$

Due to the orderly, orthogonal assembly of the confusion matrix, the relevant classification information is neatly arranged where diagonal elements contain the counts of properly classified observations while off diagonal elements give not only the count, but location of the misclassification. Foreshadowing, it is therefore wise for any measure constructed on such a matrix to utilize on and off diagonal knowledge. The results of our construction are given in Table 3.1 as an example of a 3x3 confusion matrix.

3.2.1 Definition

For any C^k confusion matrix, Class Balance Accuracy is defined as

$$CBA = \frac{\sum_{i}^{k} \frac{c_{ii}}{max(c_{i\cdot}, c_{\cdot i})}}{k} \tag{3.4}$$

		Predicted			
		Class 1	Class 2	Class 3	Total
Actual	Class 1	c_{11}	c_{12}	c_{13}	$c_{1\cdot}$
	Class 2	c_{21}	c_{22}	c_{23}	$c_{2\cdot}$
	Class 3	c_{31}	c_{32}	c_{33}	$c_{3\cdot}$
	Total	$c_{\cdot 1}$	$c_{\cdot 2}$	$c_{\cdot 3}$	N

Table 3.1 A 3x3 Confusion Matrix denoted as C^3.

A deconstruction of the above simplifies into:

$$CBA_i = \frac{c_{ii}}{max(c_{i\cdot}, c_{\cdot i})} \tag{3.5a}$$

$$CBA = \frac{\sum_{i}^{k} CBA_i}{k} \tag{3.5b}$$

A high level view of Class Balance Accuracy's construction is given in Eq. 3.4 where CBA is expressed as an overall accuracy measure built from an aggregation of individual class assessments. Individual accuracy assessments are calculated then normalized by the number of classes extant. These elements, which form the basis for the numerator, are expressed in Eq 3.5a. Information on the number of correctly predicted cases, contained in the diagonal elements, is normalized by either the total number of observations predicted to the class or the actual number of observations in that class, decided by the two greater of the two.

From its construction, CBA utilizes three core elements from each class within the contingency table: the total number of correctly classified cases, the total number of cases predicted into that class, and the total number observed in the data. Intuitively, for each class the off-diagonal row and column elements are reduced into a single sum. These singular sums form the basis for the denominator of the per class accuracy contributions. At the bottom of each per class ratio, the maximum of the row or column sum is chosen resulting in either the Recall or Precision to be the estimate of class accuracy. As a consequence, selecting the larger of the two as the denominator provides the most conservative estimate of accuracy that can be achieved. For each class, the per class Recall or Precision are aggregated and treated as the numerator

for the final ratio calculation. By using the total number of classes in the dataset as the divisor in the calculation we guarantee equal weight contributions for all classes. In the end, Class Balance Accuracy acts as a measure that independently accounts for the ability of the model to precisely recall observations from each group within the target variable.

3.2.2 Interpretation and Proof

Intuitively, as a measure, Class Balance Accuracy seeks to balance the Precision and Recall for each input class. When there is an imbalance between the Precision or Recall, a conservative process is employed such that the lower of the two measures is selected as the representative of that class's accuracy. Indeed, as the accuracy across each class is calculated, the definition of model error for any given class could be based on the model's inability to recall members of the class or overly imprecise predictions. The calculations for each class maintain their interpretations, however once averaged, the understanding that the measure provides becomes less lucid. Despite this, class balance accuracy does maintain a reasonably simple and clear meaning as a performance guarantee measure. In this capacity, class balance accuracy serves as a threshold for which the average recall and average precision of a model will not breach below. This assertion is established by the following claim.

Definition Define the following alternative forms for Class Balance Accuracy, Average Recall, and Average Precision respectfully as,

$$CBA = \frac{\sum_{i}^{k} \frac{c_{ii}}{max(c_{i\cdot}, c_{\cdot i})}}{k} \quad \& \quad \bar{R} = \frac{\sum_{i}^{k} \frac{c_{ii}}{c_{i\cdot}}}{k} \quad \& \quad \bar{P} = \frac{\sum_{i}^{k} \frac{c_{ii}}{c_{\cdot i}}}{k}$$

Theorem 3.2.1 $CBA \leq min(\bar{R}, \bar{P})$

Proof By definition,

$$c_{i\cdot} \leq max(c_{i\cdot}, c_{\cdot i}) \quad \& \quad c_{\cdot i} \leq max(c_{i\cdot}, c_{\cdot i})$$

Taking the reciprocal and dividing by c_{ii},

$$\frac{c_{ii}}{c_{i\cdot}} \geq \frac{c_{ii}}{max(c_{i\cdot}, c_{\cdot i})} \quad \& \quad \frac{c_{ii}}{c_{\cdot i}} \geq \frac{c_{ii}}{max(c_{i\cdot}, c_{\cdot i})}$$

Summing across all classes and dividing by the number of groups yields,

$$\frac{\sum_{i}^{k} \frac{c_{ii}}{c_{i.}}}{k} \geq \frac{\sum_{i}^{k} \frac{c_{ii}}{max(c_{i.},c_{.i})}}{k} \quad \& \quad \frac{\sum_{i}^{k} \frac{c_{ii}}{c_{.i}}}{k} \geq \frac{\sum_{i}^{k} \frac{c_{ii}}{max(c_{i.},c_{.i})}}{k}$$

By definition,

$$\bar{R} \geq CBA \quad \& \quad \bar{P} \geq CBA$$

Therefore, by the identity property of minimums,

$$CBA = min(CBA, CBA) \leq min(\bar{R}, \bar{P}) \quad \blacksquare$$

The proof of the claim begins with a statement that each per class row and columns sum are less than or equal to the maximum of those two numbers. Each side is then divided by the total number of correct observations, and the reciprocal is taken. This simultaneously reverses the inequality and defines the per class recall and precision contributions. It is at this point in the proofs development that the implication is obvious. On the right side of each inequality, Class Balance Accuracy selects the measure with the lowest value as the representative accuracy. In doing so it creates a conservative estimate of that class's contribution to the overall accuracy. To complete the proof, we sum across the k number of groups and then divide by said number. With the complete definitions of average precision and average recall, the relationship shows that each measure will be greater than or equal to the class balance accuracy value. To combine these two separate inequalities the identity property of minimums was used to show that the smallest of the average precision and average recall will be greater than or equal to the Class Balance Accuracy. Proof for the measures interpretation shows that for any kxk confusion matrix, Class Balance Accuracy is a simultaneous lower bound for both the average Recall and average Precision. Therefore, Class Balance Accuracy can be interpreted as a performance guarantee metric where the average precision and average recall for a model are bounded below by the calculated CBA value. As an evaluation tool, CBA creates an overall assessment of model predictive power by scrutinizing measures simultaneously across each class in a conservative manner that guarantees that a model's ability to recall observations from each class and its ability to do so efficiently won't fall below the bound. We also state, without proof, that

class balance accuracy the greatest lower bound for the average precision and average recall across each group. All things considered, as a multi-class measure it accounts for overall in her class performance in a conservative and intuitive manner.

As a more stylized discussion of CBA's classification assessment, the reader may imagine a family of acrobats who specialize in high risk tight rope acts. It is clear the success of each individual is integral to the preservation and happiness of the group as a whole, hence we must account for each member independently. That said, as a unit they must all perform well for the show to be a triumph. So at any given performance, one tight rope houses all members, each individually attempting to stay at equilibrium as they walk across. This is akin to the classifier recalling as many observations as possible from a given class and doing so with a high level of precision, effectively balancing these two equally important metrics. As each member attempts to walk across carefully, and the classifier analogously attempts to group observations into each class, Class Balance Accuracy will ultimately rank the classifier highly if it can enable each individual to make it across while keeping both sides of the beam balanced. Some classes, often the majority ones, will be biased towards higher recall and low precisions, while minority ones are likely to suffer from the opposite effect of low recall, but high precision. This conceptualization highlights the fact that each per class accuracy contribution can be represented by a left leaning recall deficient or a right tilted precision problem. When viewed as a whole, each member is slightly tilted in different directions, but all are working towards the singular goal of making it across safely. By accounting for the effect of each class, CBA contrasts with traditional accuracy measures that simply attempt to ensure the "most important" family member makes it across, notwithstanding and possibly to the detriment of everyone else.

3.2.3 Properties

For further investigation into the properties of class balance accuracy, we must relate how it's functional form translates the information found in contingency matrices into error estimates under a variety of scenarios. However, before that discussion can begin, a few concepts must

be introduced.

Definition *Discriminancy* For two measures f and g on domain ψ, let $P = \{(a,b)|a,b \in \psi, f(a) > f(b), g(a) = g(b)\}$ and $Q = \{(a,b)|a,b \in \psi, g(a) > g(b), f(a) = f(b)\}$. The degree of discriminancy for f over g is $D = \frac{|P|}{|Q|}$.

Defined by Huang and Ling, discriminancy and consistency can be used to compare how to measures evaluate information. Discriminancy quantifies the differences in range between two measures as a ratio of the total number of possible output values of the two measures. Specifically applied to contingency table analysis, one measure has more discriminancy over another measure when it's range of values over the same set of contingency tables is larger. The following figure elucidates the definition of discriminancy for five matrix reduction based measures: Balanced Accuracy, Regular Accuracy, Class Balance, G-Mean, and F-Score. In figure 3.2, the five constructed matrices are hypothetical representations of the predictions of five modeling outputs. Summing across the rows informs us that there are 50 observations in class 1 and 100 observations in class 2, which yields an imbalance ratio of 2 to 1, majority to minority. Under each table, we have the value of the measure calculated from the table above. Hence for the first table, the Balanced Accuracy is equal to 60% while the F-Score for the same table is 44.4%. From the measure output values for each table, we can directly assess the degree of discriminancy for each measure. Balance Accuracy, which accounts for the recall over each class has the weakest ability to discriminate between different contingency matrix inputs and of the five matrices it can return only two distinct values. Three of the five metrics, Regular Accuracy, G mean, and F-score are able to return for distinct values across five different matrices. It is only Class Balance Accuracy's ability to account for both row and column sum simultaneously that allows it to discriminate between all five matrices.

Definition *Consistency* For two measures f and g on domain ψ, letR $= \{$(a,b)|a,b $\in \psi, f(a) > f(b), g(a) > g(b)\}$ and $S = \{(a,b)|a,b \in \psi, f(a) > f(b), g(a) < g(b)\}$. The degree of discriminancy for f and g is $C = \frac{|R|}{|R|+|S|}$, where $0 \leq C \leq 1$.

Measure consistency describes to what degree two evaluation metrics move in tandem across different inputs. Figure 3.2 highlights that difference between measure values when calculat-

	C_1	C_2
C_1	50	0
C_2	80	20

C_a^2

	C_1	C_2
C_1	30	20
C_2	40	60

C_b^2

	C_1	C_2
C_1	40	10
C_2	60	40

C_c^2

	C_1	C_2
C_1	10	40
C_2	0	100

C_d^2

	C_1	C_2
C_1	50	0
C_2	70	30

C_e^2

	C_a^2	C_b^2	C_c^2	C_d^2	C_e^2
Balanced Accuracy	.600	.600	.600	.600	.650
Regular Accuracy	.466	.600	.533	.733	.533
Class Balance	.292	.514	.400	.457	.358
G-Mean	.447	.600	.566	.447	.548
F-Score	.444	.583	.533	.583	.525

Figure 3.2 Values for five metrics across five matrices. CBA has the strongest discriminancy ability, returning five distinct values, one for each matrix.

ing the third and fourth matrices is positive for three of the five measures. Balance accuracy cannot discriminate and has no change, while the geometric mean actually decreases. In this scenario, regular accuracy, Class Balance Accuracy and F-Score all exhibit consistency with one another. A comparison between the predictions between matrices C_c^2 and C_d^2 point to a difference in the precision of the minority class predictions and the mount of total recall for said class. The level of consistency for these measures indicate that each ranks models higher that can predict minority classes with high precision over the indiscriminate allocation of majority class observations into the minority group.

When taken together, the degree of discriminancy of a measure will determine the cardinality of its range which directly relates to, but is not a function of, how consistent the measure is with other metrics. Ultimately, both properties dictate how a measure will rank order contingency matrices and it should be noted that the differences between the measures is largely an effect of how off-diagonal information is processed. It is the exploration of how information is processed that motivates the use of the measure evaluation taxonomy developed by Sokolova and Lapalme in the following section.

The measure evaluation taxonomy is a structured framework for understanding when the eight suggested invariance properties were tested for CBA and the other three competing multi-class measures. From a high level standpoint, these invariance properties can offer a quick view into how a measures is processing on and off diagonal information. Table 3.2 is a recreation of Sokolova's results. Given the similarity of its pattern to the other measures in its class, the validity of Class Balance Accuracy as a bona fide, unique accuracy measure should become more apparent. CBA's construction allows it to process information in a similar fashion to the other, more complex information theoretic measures.

As a short walkthrough, Sokolova's properties will be discussed within the context of Class Balance Accuracy. The first invariance property introduced is one that tests whether a measure is invariant under an exchange of positive and negative classes. This is akin to simply switching the labels, and it should only be expected for a measure to be invariant to the class name. Properties 2 and 3 describe scenarios where the true positive or true negative counts are

changed. Naturally any valid measure should be able to detect a change in either of the true counts. As so, CBA is non-invariant in this case. Properties 4 and 5 describe changes in the false negative and positive counts. Here Class Balance Accuracy is quasi-invariant to changes on the off diagonal elements since the accuracy value doesn't change until the direction of the difference between the row or column sum changes. Balance between Precision and Recall are forthright, CBA is not interested in the specific counts within each off-diagonal cell. The other multi-class measures explicitly take into account these counts, which add to their discriminatory power. The last three properties all assess a measure's ability to deal with multiplicative changes in sample size, either uniformly, by row or by column. CBA naturally quantifies this information, and as Sokolova et.al. point out, invariance on these properties suggest a measures ability to assess performance on different classes.

Invariance Property	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
Binary Classification								
Accuracy	-	Δ	Δ	Δ	Δ	-	Δ	Δ
Precision	Δ	-	Δ	-	Δ	-	-	Δ
Recall (Sensitivity)	Δ	-	Δ	Δ	-	-	Δ	-
Fscore	Δ	-	Δ	Δ	Δ	-	Δ	Δ
Specificity	Δ	Δ	-	-	Δ	-	Δ	-
AUC	Δ	Δ	Δ	Δ	Δ	-	Δ	-
MultiΔclass Classification								
CBA	-	Δ	Δ	$\pm$	$\pm$	-	Δ	Δ
MCC	-	Δ	Δ	Δ	Δ	-	Δ	Δ
CEN	-	Δ	Δ	Δ	Δ	-	Δ	Δ
RCI	-	Δ	Δ	Δ	Δ	-	Δ	Δ

Table 3.2 Invariance properties for performance criteria across binary and multi-class classification tasks. Let "-" represent invariance, "Δ" denote non-invariance and "$\pm$" highlight quasi-invariance.

Class Balance Accuracy shares many of the invariance properties of other multiclass performance metrics, which besides overall accuracy are all information theory based. CBA's ability to detect changes in the false negative and false positive counts separate it from overall accuracy. This table highlights that despite being a matrix reduction technique, which have been historically less complex than information theory base metrics, class balance accuracy achieves a comparable level of discriminancy while remaining simple and intuitive.

3.3 Calculation Examples

At this point, through example we would like to begin bridging Class Balance Accuracy's theoretical construction and practical application. The hope is to provide meaningful 2x2 and 3x3 confusion matrix examples that will help solidify CBA's validity and aid in the interpretation of its values. The discussion will begin with a comparison of Class Balance Accuracy and Regular Accuracy for the 2x2 case, and proceed to compare its calculations against the multi-class measures for the 3x3 instance.

3.3.0.1 Comparison between Accuracy and Class Balance Accuracy under 2x2 Class Imbalance

Consider the standard 2x2 matrices displayed in Table 3.3. Summing across the rows, note Class 1 as the majority group with 60 observations and Class 2 is the minority group with only 10 observations. The generating classifier was only able to successfully classify Class 1 cases. This will be our reference matrix as we see how Class Balance Accuracy changes as observations are correctly predicted into the minority class.

(a)

	C_1	C_2
C_1	40	20
C_2	10	0

(b)

	C_1	C_2
C_1	41	19
C_2	10	0

(c)

	C_1	C_2
C_1	40	20
C_2	9	1

Table 3.3 2x2 Confusion matrices highlighting the change in accuracies as minority or majority classes are correctly classified.

Regular Accuracy, as calculated from Table 3.3(a), is .571. The Class Balance Accuracy is .333 as derived from averaging the sum of 40/60 and 0/20. From here, let's observe how the

values of Class Balance Accuracy vary as the number of incorrectly classified cases diminishes. In Table 3.3(b), an erstwhile false negative prediction is correctly classified as a true positive, a recall increase. CBA recognizes the additional accuracy and returns a value of .341. At this point, no Class 2 cases have been predicted properly in either Table 3.3(a) or 3.3(b). Table 3.3(c) displays the change where a minority class observation is correctly assigned. Subsequently, the matrix as a whole receives a higher Class Balance Accuracy score, .357. The main result is that CBA values a classifier's devotion to minority class prediction over increases in additional majority recall. An analogous restatement is, in the presence of class imbalance, majority class precision is deemed more important because it portends to an increase in minority class recall.

3.3.0.2 Comparisons between Multi-class Measures with and without Class Imbalance

As both a binary and multi-class measure, Class Balance Accuracy can be examined for confusion matrices beyond $k = 2$. We will now present various balanced and unbalanced special cases to gain intuitive insight into Class Balance Accuracy values as compared to the other multi-class measures. This will serve as a primer for the following section where the measures will be used in practice for tasks such as model assessment.

(a)

	C_1	C_2	C_3
C_1	33	33	33
C_2	33	33	33
C_2	33	33	33

Table 3.4 Special case 3x3 confusion matrices without class imbalance where all cells are equal.

(b)

	C_1	C_2	C_3
C_1	99	0	0
C_2	99	0	0
C_2	99	0	0

Table 3.5 Special case 3x3 confusion matrices without class imbalance where all observations have been predicted into one class.

(c)

	C_1	C_2	C_3
C_1	99	0	0
C_2	0	99	0
C_2	0	0	99

Table 3.6 Special case 3x3 confusion matrices without class imbalance where each class has been perfectly classified.

(d)

	C_1	C_2	C_3
C_1	0	0	99
C_2	0	99	0
C_2	99	0	0

Table 3.7 Special case 3x3 confusion matrices without class imbalance where one class is perfectly classified and all other observations have their labels switched by the classifier.

Measure	C_a^3	C_b^3	C_c^3	C_d^3
MCC	.000	.000	1.000	.000
CEN	.861	.333	.000	.333
RCI	.000	.000	1.000	1.000
CBA	.333	.111	1.000	.333
RA	.333	.333	1.000	.333

Table 3.8 Measure values calculated from Table 3.3 through Table 3.7.

Values across the various measures, as seen in Table 3.8. are generally standard. Confusion Entropy's value of .861 appears as an oddling, but simply represents a high amount of information loss, and is consistent with the other measures. CBA returns a value of .11 for Table 3.6(b), which is the lowest among the three tables. Despite the uniform assignment of the classifier, Class Balance Accuracy respects that even this arbitrary assignment of classes does have some predictive power. The Relative Classifier Information values for both 3.6(c) and 3.7(d) should immediately be alarming. This phenomena, like all measure peculiarities, is

a consequence of its construction where significance is placed on the overlap of the input and output densities.

Continuing on to Tables 3.9 - 3.12, Table 3.9. describes the multi-class measures in the presence of imbalance. Each scenario is quite uncommon but important for understanding how the multi-class characterize algorithm performance. For these examples, the distributions of the groups are skewed towards Class 1. In the first two tables, one group is perfectly classified while the others are perfectly misspecified. The differentiating feature is whether the perfectly classified class was a majority or minority group. Tables 3.11(c) and 3.12(d) contain the confusion matrices for random assignments based on partitioning the data. Table 3.11(c) splits the data into thirds and label arbitrarily assigns a class label. In the last example, one can imagine a scheme where the classifier simply takes the given class proportions and randomly assigns labels according to this prior probability. Matthew's Correlation Coefficient returns its highest value for matrix C_a^3 and interprets this situation more favorably than all others. These examples highlight the strength of MCC. As a measure it can correctly identify random assignments of data with more consistency than the other measures. CEN performs well due to its discriminatory power, however it fails to recognize the randomness in Table 3.11(c), despite the off diagonal assignments being a clue that the classifier isn't performing as it should. As seen previously, RCI is looking for distinction between groups, and largely ignores the actual operational environment. Class Balance Accuracy, as a per class measures, gives equal weight to both classifiers used to derive Tables 3.9(a) and 3.10(b). The perfectly classified class is contributing its maximum allotment to the measure, while all other classes contribute zero, hence the $1/k$ value. In the third table, CBA recognizes the lack of recall, and punishes this classifier accordingly. Similarly, because randomly assigning classes based on proportions will produce confusion matrices with skewed structures, CBA again weights the lack of recall and precision though they are equal for each class.

In conclusion, these results through example highlight the well-established fact that different classifiers will not rank order the classifiers identically and when assessing models the objective is an important consideration (Nguyen et.al, 2009). Furthermore as suggested by Baldi et.al, the measures construction is important to understanding how a measure will perform

in practice and it is often necessary to list or combine measures to get general understanding of a classifiers properties. Of the multi-class measures listed, MCC is best reserved for understanding if a classifier is randomly assigning class labels. CEN can be used for situations where discrimination between confusion matrices is important. RCI is important for ranking uniformity of predictions, while willfully ignoring if the classes have been predicted correctly or not. Regular Accuracy is still the best method for determining the number of correctly classified observations. Class Balance Accuracy now has its own unique scenario for use. When algorithm performance across each class is a focal point, Class Balance Accuracy should be used to discriminate between techniques that focus on a observations from a majority class at the expense of minority cases.

(a)

	C_1	C_2	C_3
C_1	170	0	0
C_2	0	0	20
C_2	0	10	0

Table 3.9 The majority class is perfectly predicted and no others.

(b)

	C_1	C_2	C_3
C_1	0	0	170
C_2	0	20	0
C_2	10	0	0

Table 3.10 A minority class is perfectly predicted.

(c)

	C_1	C_2	C_3
C_1	57	57	56
C_2	6	7	7
C_2	3	3	4

Table 3.11 One third of the cases are randomly assigned to each group.

(d)

	C_1	C_2	C_3
C_1	145	17	8
C_2	17	2	1
C_2	8	1	1

Table 3.12 Observations are assigned to classes based on the natural proportion of the data.

Measure	C_a^3	C_b^3	C_c^3	C_d^3
MCC	.443	.018	.011	.019
CEN	.069	.139	.574	.395
RCI	1.000	1.000	.001	.002
CBA	.333	.333	.162	.351
RA	.85	.100	.34	.74

Table 3.13 Multi-class measure values for each instance.

3.4 On the Use of Class Balance Accuracy in Controlled and Uncontrolled Environments

To investigate the use of Class Balance Accuracy in practice, both investigative and controlled simulated studies were arranged to help garner more insight into its performance as a model evaluation tool. The first of these studies was designed simply to assess the characteristics of models chosen when several measures were maximized. Results from this investigative study motivated a more formalized simulation experiment that sought to gain an understanding of situations when CBA will outperform Regular Accuracy. Lastly, with the final group of simulations, we further compare measure performance when selecting models trained with varying amounts of data. Altogether these studies will show how class balance accuracy performs as a model performance metric by viewing its characteristics from both a theoretical and practical perspective.

3.4.1 Study 1: Initial Investigations into Class Balance Accuracy's Practical Application

In practice, to get an understanding of model performance, measurement values are calculated from the final predictions.it is the intent of the measure to discriminate between models according to their performance as defined by some objective. This objective could be how

well the model performs overall, how well the model performs for each class, or even when a model makes a prediction how often is that prediction correct. Each one of these is a different perspective for which a model can be critiqued, scrutinizing between models that fulfill or fail to meet the objective. In the previous chapter we discussed and have shown how different measures viewed the performance of models according to final output values for various prediction scenarios. Though the ultimate goal is to select the model that can make quality predictions robustly beyond just the data observed, it prudent of us to understand that though we often don't necessarily view measures as being different perspective of model performance, they do and by their different constructions each synthesize and highlight different aspect of the predicted results. As a first look into this, for each of the data sets, six models; Naive Bayes, Classification Trees, Neural Networks, Support Vector Machines, Linear Discriminant Analysis and Random Forests, were fitted to the full data set. Using the output predictions, seven performance metrics were calculated and for each metric every model was ranked. The top performing model for each measure was returned and the statistics around those calculations were recorded. This process resulted in a total of 96 model runs which corresponded to 672 performance computations. To facilitate our discussion, we will view examples that highlight the differences between models chosen by CBA and other measures.

Measure	Choice Model	Groups Predicted	Accuracy	Counts
cba	bayes	23 of 24	0.588	133
fscore	bayes	23 of 24	0.588	133
gmean	tree	6 of 24	0.487	110
ba	bayes	23 of 24	0.588	133
mcc	nnet	20 of 24	0.695	157
cen	forest	8 of 24	0.434	98
oa	nnet	20 of 24	0.695	157

Table 3.14 Top performing models for each performance metric as assessed after training on the full Audio dataset.

When viewing these results, we will take into consideration the overall accuracy and the number of classes predicted under each learned model. For the Audio data set, a total of four distinct models were chosen across the seven metrics, with two of those models drastically underperforming the others. Due to the structure of the data set simple rule based partition

methods are insufficient in modeling the data space. Neither random forest or classification trees achieved a level of accuracy above 50%. Both of these models also performed poorly across the classes. We now come to the divergence between the model selection criteria. Models chosen by maximizing Matthews correlation coefficient and overall accuracy had only a 30% overall error rate, the lowest of all the models. A side effect of selecting bounds that maximize the overall accuracy, we have sacrificed the ability to predict three of the 24 classes while other models are able to account for these groups. The naive Bayes model, as chosen by maximizing Balanced Accuracy, Class Balance Accuracy and Recall, had a slightly higher misclassification rate of a little over 40%, but was able to account for three of the models that the neural network technique could not. We begin to see the behavior of measures that account for classes independently. They tend to uplift models that predict well across all classes while denigrating those who can't.

Measure	Choice Model	Groups Predicted	Accuracy	Counts
cba	nnet	6 of 8	0.86	289
fscore	bayes	6 of 8	0.86	289
gmean	tree	5 of 8	0.86	289
ba	bayes	6 of 8	0.86	289
mcc	svm	5 of 8	0.869	292
cen	forest	5 of 8	0.821	276
oa	svm	5 of 8	0.869	292

Table 3.15 Top performing models for each performance metric as assessed after training on the full E. coli dataset.

For the last dataset, three of the four class independent measures returned the same model, which may suggest that they process contingency table input identically. After training on the E. coli data set and ranking the models an alternative picture emerges. Across all seven measures, six distinct models were chosen. Support vector machines the model chosen by MCC and Overall Accuracy, recalled the largest total number of correct labels yet only outpaced the other models by three total observations while failing to identify an entire group. Neural nets, naive Bayes, and classification trees where the models chosen by the class independent measures, all of which except classification trees were able to recall six of the eight classes. The difference between the models selected gives us an opportunity to gain a deeper appreciation

for the way class balance accuracy scores models. Considering the three models all have the same overall accuracy, having been able to classify 289 cases, we must look at the breakdown of the correct number of observations within each class to differentiate between the models.

	cp	im	imL	imS	imU	om	omL	pp
tree	142	71	0	0	19	15	0	42
svm	136	67	0	0	24	18	0	47
lda	140	57	0	0	26	19	3	44
bayes	135	57	0	0	31	18	4	44
forest	136	61	0	0	22	16	0	41
nnet	138	63	0	0	23	18	4	43

Table 3.16 Per class recall for the E. coli dataset.

From Table 2.2, we are reminded that the E. coli data set has 8 groups, three of which have extremely low representation. The "imL", "imS", and "omL" classes have sample sizes of 2, 2, and 5, respectively. These classes are difficult to predict for most of the classifiers and only LDA, naive Bayes, and neural networks are able to categorize any these observations. The difference between the two highest performing per class models, naive Bayes and Neural Networks, is expressed by an increase of the number of observations predicted into the CP and IM groups for neural networks, and increase in the IMU and PP groups for naive Bayes. This difference lends itself to an increase in the majority classes for the model selected by Class Balance Accuracy which by virtue of its construction is sacrificing most of his observations from the "imU" group in order to balance the recall and precision for the CP and IM classes. When analyzing the results from models fit on the nursery data set the effects of class balance accuracy's attempt to create symmetry between the average precision and recall are more pronounced. The F-score is a similar metric that accounts for the same contingency matrix characteristics as Class Balance Accuracy, however it does so by taking the harmonic mean of the two. For this particular data set it selects random forests, the same model chosen when maximizing the more traditional overall performance measures. When we compare this model's performance to the performance of neural networks, as chosen by CBA, we see that the main differential is how each model treats the "priority", "spec prior" and "very recommended" classes. Neural Networks makes a sizeable trade-off in predicting the "priority" and smaller

one for the "spec prior" classes in favor of higher recall for the "very recommended" group. Quantitatively, the number of percentage point differences between the different recall values for the "priority" and "spec prior" classes are 8.6% and 2.1%, respectfully. The total difference of 10.7% is still just above half of the recall Neural Networks gain by shifting focus to the minority class "very recommended" cases. Recall gain for this class was 18.9% points when using Neural Networks for the classifier. Class balance accuracy prefers to select what may be dubbed "Caste-Free" models, ones willing to sacrifice the performance of any one class for the scale of overall class performance. Because of the nature of class imbalance, this sacrifice is often made at the expense of the majority class towards underrepresented groups though the measure does account for the level of trade-off between predicting observations of these two groups and in most cases prevents majority class observations from shouldering the full burden of precision and/or recall.

Measure	Choice Model	Groups Predicted	Accuracy	Counts
cba	nnet	4 of 5	0.947	12272
fscore	forest	4 of 5	0.977	12666
gmean	tree	3 of 5	0.873	11310
ba	nnet	4 of 5	0.947	12272
mcc	forest	4 of 5	0.977	12666
cen	lda	4 of 5	0.548	7107
oa	forest	4 of 5	0.977	12666

Table 3.17 Top performing models for each performance metric as assessed after training on the full Nursery dataset.

	not recom	priority	recommend	spec prior	very recom
tree	4320	3324	0	3666	0
svm	4320	4152	0	3994	199
lda	1257	2980	0	2788	82
bayes	4320	3852	0	3512	20
forest	4320	4147	0	3999	200
nnet	4320	3778	0	3912	262

Table 3.18 Per class recall for the Nursery dataset.

For the following tables, the rankings of each model selected by the measure are organized by data set. We look at both perspectives, per class and overall, to determine how well the

model selected by each measure compare with one another. If we allow the object is to maximize overall accuracy, selecting Matthew's Correlation Coefficient and Regular Accuracy, we consistently select the models that perform the best. Both measures are consistent across each data set. Of the independent class measures, the F-Score ranked the best, beating out Class Balance Accuracy on the Optidigits data set to achieve the top position. It is encouraging that when considering all classes, we still are able to select models that perform well in the aggregate, as it would be discouraging to maximize on the micro-scale and do poorly on the macro-scale.

As we shift our objective to per class measurement, the independent class measures performance shines. CBA, F-Score and Balance Accuracy all consistently choose the models that have the best per class accuracy. Here we may note that the use of G-Mean as an independent measure per class accuracy underperforms both of the overall accuracy measures. Interestingly, G-Mean has been suggested in the class imbalance and some instance selection literature papers as a suitable alterative for measuring per class accuracy over regular accuracy. our results show that by maximizing the overall accuracy we select models that perform better per class than models chosen by the geometric mean.

3.4.2 Study 2: All-Red Boundary Tests

In light of the results in the previous section, we find that maximizing overall accuracy and per class accuracy can select models that perform similarly. Maximizing class balance accuracy, consistently selected the best models for the per class objective and as expected, maximizing overall accuracy resulted in models that achieve the highest total number of correct observations, yet both performed reasonably well at the other's natural objective. It became the author's curiosity to delve deeper into these differences by designing a simulation study that will compare measure performance as a function of a few key criteria that are extant in class imbalance. With this in mind, we chose to explore how well class balance accuracy and regular accuracy could differentiate between a straw-man model and one derived from the true bounds. In previous chapters, we noted that the lack of data and high degrees of concept complexity

	Anneal	Audio	Bal.	Ecoli	Flare	Glass	Hep.	Nur.	Opti	Page	Pen.	Sat.	Seg.	Soy	Yeasts	Dia.
cba	2	2	1	2	1	1	1	2	1	1	1	1	1	1	1	2
fscore	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1	2
gmean	2	4	1	2	1	1	2	3	1	1	1	1	1	2	2	2
ba	2	2	1	2	1	1	2	2	1	1	1	1	1	1	1	2
mcc	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
cen	3	4	2	3	2	2	2	4	2	2	2	2	2	2	3	3
oa	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 3.19 Measure rankings according to overall performance.

	Anneal	Audio	Bal.	Ecoli	Flare	Glass	Hep.	Nur.	Opti	Page	Pen.	Sat.	Seg.	Soy	Yeasts	Dia.
cba	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
fscore	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gmean	1	4	1	2	1	1	1	2	1	1	1	1	1	2	2	1
ba	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
mcc	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1
cen	3	3	2	2	1	1	1	1	2	1	1	1	2	2	1	1
oa	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1

Table 3.20 Measure rankings according to per class performance.

are two aspects of the class imbalance problem that hinder not only prediction but, model evaluation. Therefore a simulation study was designed explicitly to determine if maximizing class balance accuracy in the presence of these extreme conditions would result in the selection of the true model more often than regular accuracy. If so, it would shed more light into situations where maximizing class balance accuracy may be more beneficial for overall performance than regular accuracy itself.

In the design of experiments spirit, a two factor completely randomized factorial computer experiment was created. The two factors of interest were sample size and degree of separability. Each data set in this simulation was created by a randomly selecting a user-specified number of observations, with specific class labels, within a 100 x 100 grid. To create separation between the groups a true bound was placed at 50 units along the x-axis which subsequently divided the grid into two halves. The left half of the grid and would contain predominantly red observations and the right side predominantly green, with overlap allowed. This true bound served as one model. The straw man alternative would be an all-red model that predicts every observation into the red class. This model is not only weak for its predictive power, but it's explanatory ability as well because its blanket predictions add no new information. For class imbalance problems, models such as these are the bane of researchers, because in some instances they will return very high levels of accuracy only to contain no value added as the practitioner herself could have made a classification ruled that assumes all observations are of the majority class. Continuing on with the design, to emulate separability, the ratio of red to green observations were adjusted within each half. For example, the high separability level within the concept complexity factor would have any initial data set that consists 20,000 red observations and 1,000 green observations on the left side and 1,000 red observations and 20,000 green observations on the right. Our concept complexity factor, in all, contained four levels. To evaluate the effect of sample size, twelve levels ranging from 5 data points up to 500 would be randomly selected from the whole data set. In all, this factorial design tested 48 different combinations. For each combination, 1000 repetitions were run, bringing the total number of simulations to 48,000. Each iteration within each combination represents a different data set, and hence on each repetition we calculated the class balance accuracy of the true

model and the all-red model, along with the regular accuracy of the two. For each measure, like done previously, we chose the model with the highest measured value for each metric. We then categorized the results into one of seven categories that checked whether: "Both Models were Incorrect", "Only CBA chose the Correct Model", "Only RA chose the Correct Model", "Both chose the Correct Model", "Neither could Differentiate between the Models", "RA could not Differentiate between the models & CBA choose the Incorrect model", "RA could not Differentiate between the models & CBA choose the Correct model", "CBA could not Differentiate between the models & RA choose the Incorrect model", "CBA could not Differentiate between the models & RA choose the Correct model". After accounting for all of these scenarios across each iteration, the proportions were plotted on a line graph where the x-axis contains the sample size and the y-axis the average proportion of each simulation outcome. For each level of concept complexity there is a separate plot displaying the convergence curves.

We see from the figures that the results are quite intuitive. For the high separability case, there is a fast rate of convergence where both measures select the true model 100% of the time. Looking above, this high level of separability allows for the easy delineation of the red and green classes. At extremely low sample sizes we do see some deviation from perfect selection, but it is in no way pronounced. Moving to the average separability level, we see the ratio of green to red observations on the right-hand side decrease allowing us to see visually that there is less separation between the two groups. Looking at the convergence curve, even for the lowest amount of data, both measures are correct about 77% of the time. The next largest category of simulation outcomes is where neither model could differentiate between models. Looking at the results under partial separability, the trend becomes more pronounced and we begin to see more diversity in the simulation outcomes for the smaller sample sizes.

Having now made it to the low separability results, let us take our time to synthesize the outcome. First looking at the data, we see that the two groups are barely distinguished by the bounds. There is such a high level of imbalance that even visually partitioning the groups would be tough without knowing the model for which the data was simulated from. Of all the levels, the convergence curve corresponding to high concept complexity shows the most diversity of the four levels. The same overall trend occurs, such that as we increase the sample size both

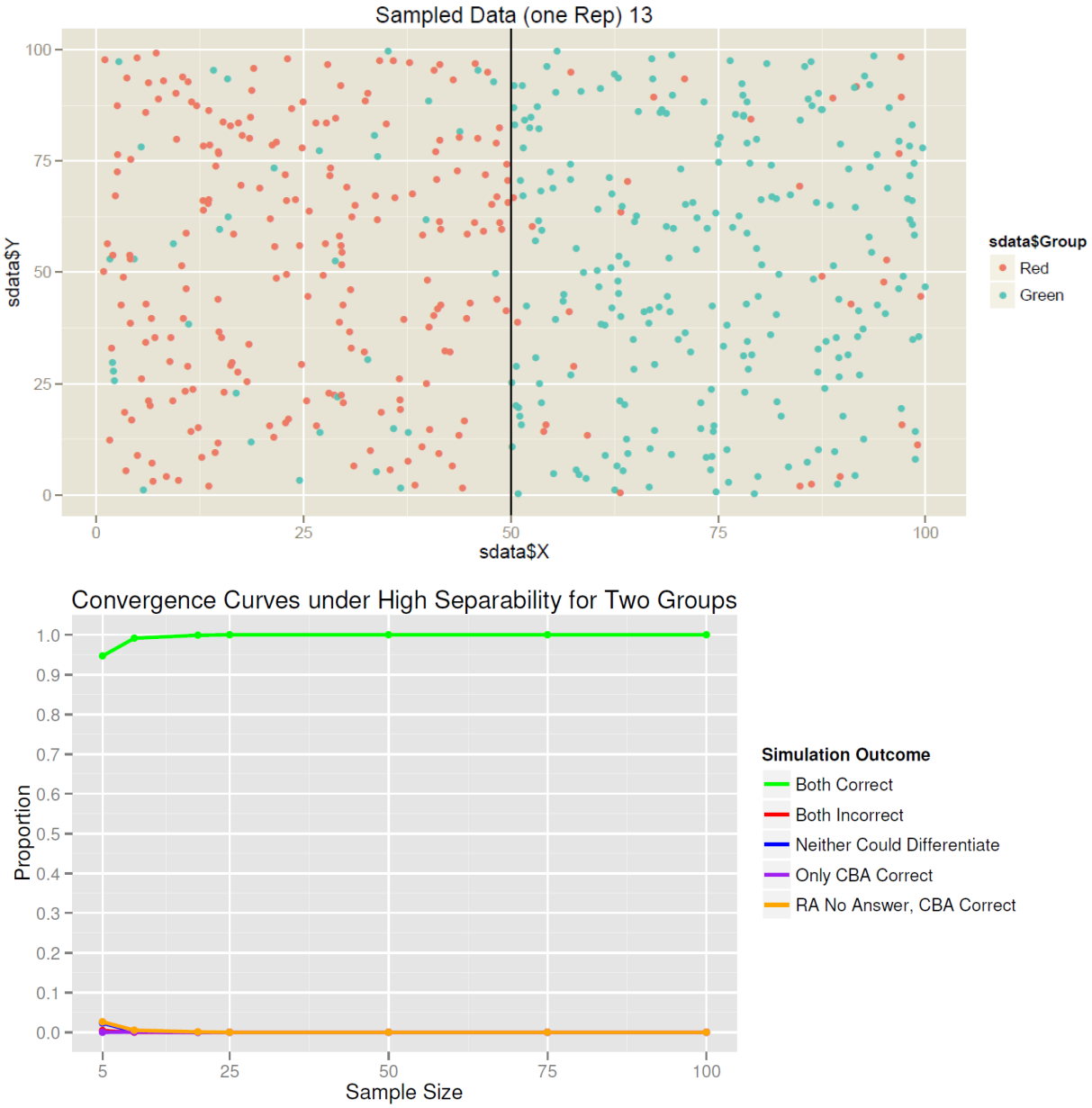

Figure 3.3 Data snapshot and convergence curves for two groups in a highly separable scenario.

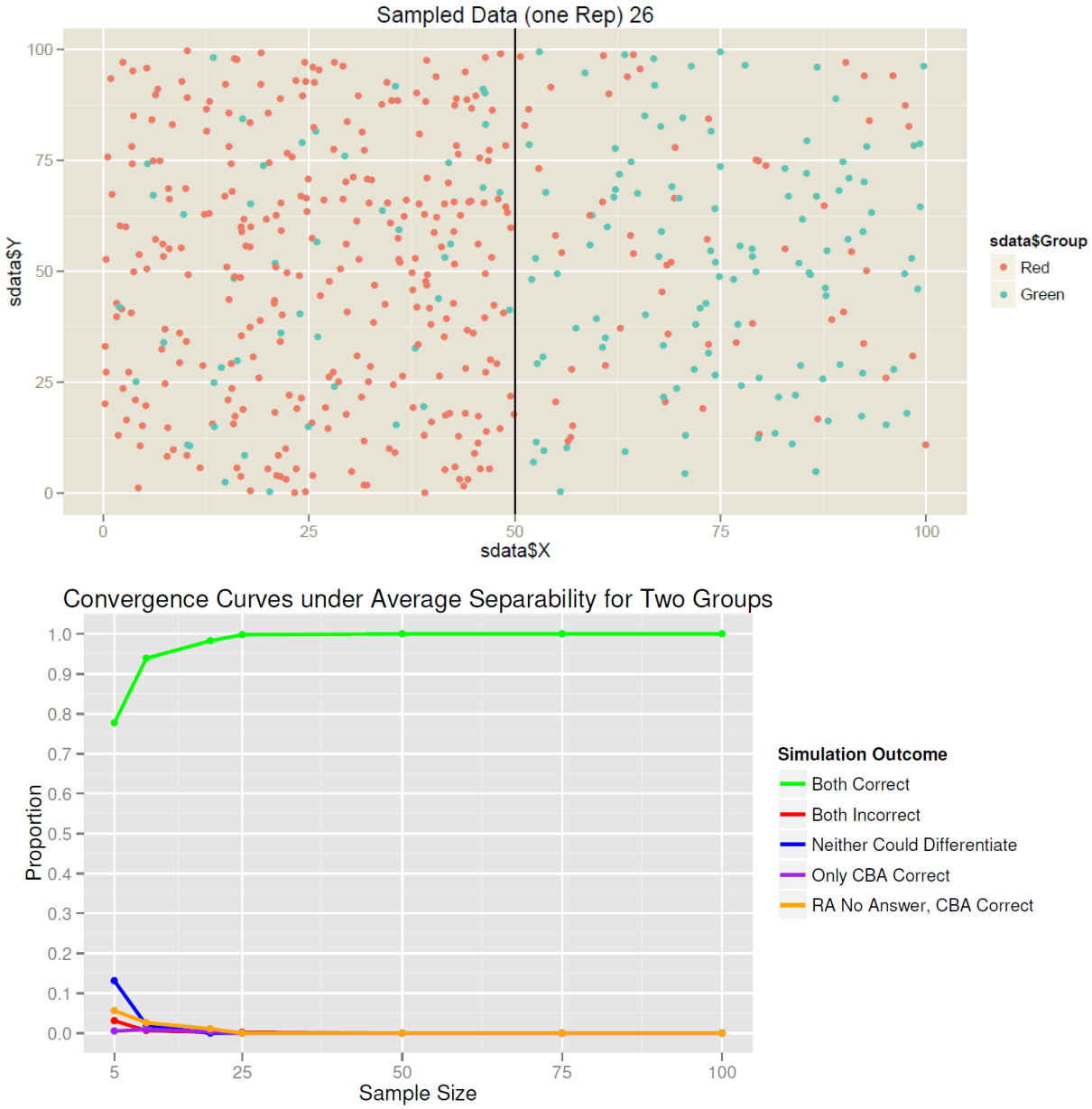

Figure 3.4 Data snapshot and convergence curves for two groups in a scenario with average separability.

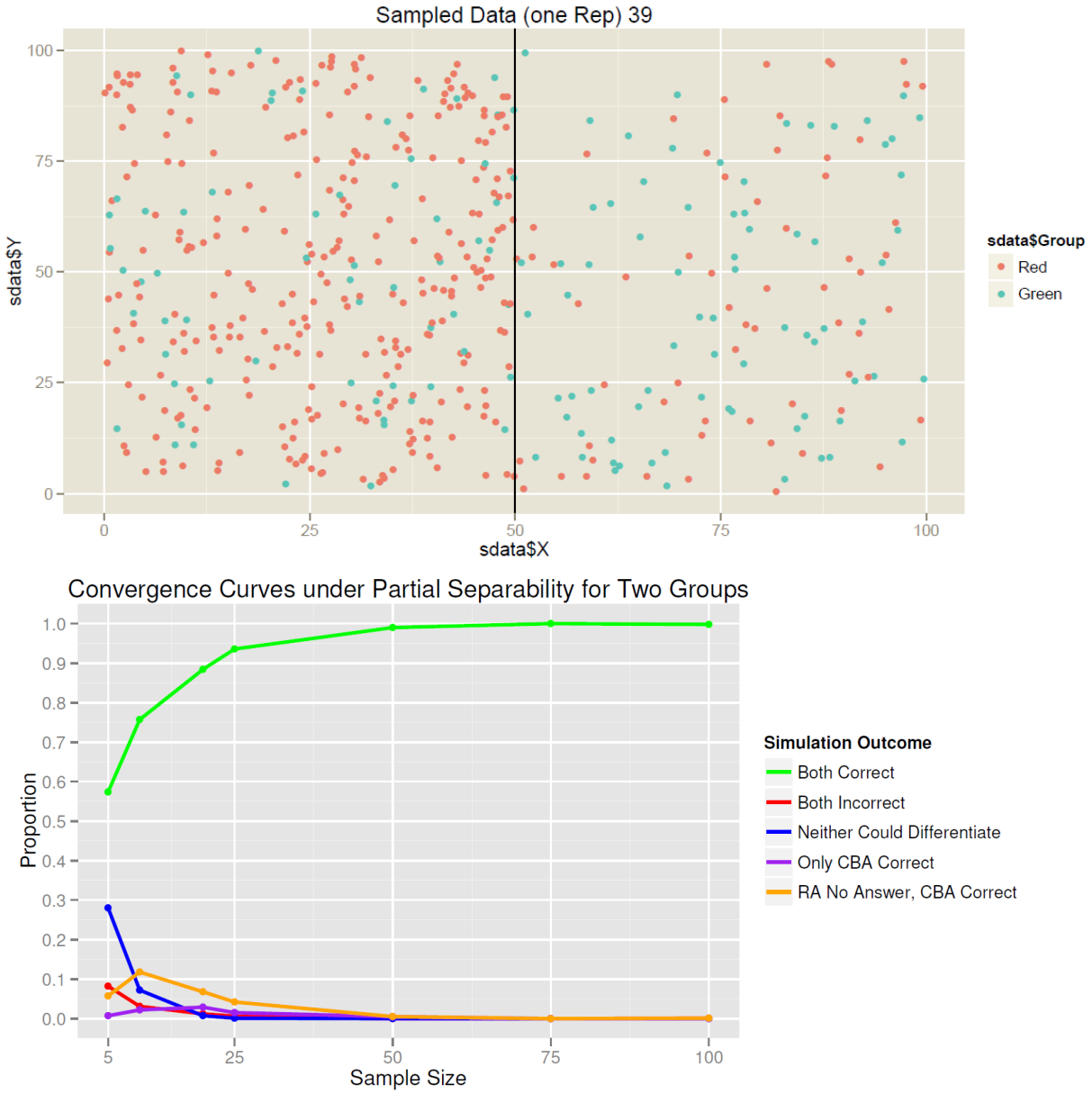

Figure 3.5 Data snapshot and convergence curves for two groups in a partially separable scenario.

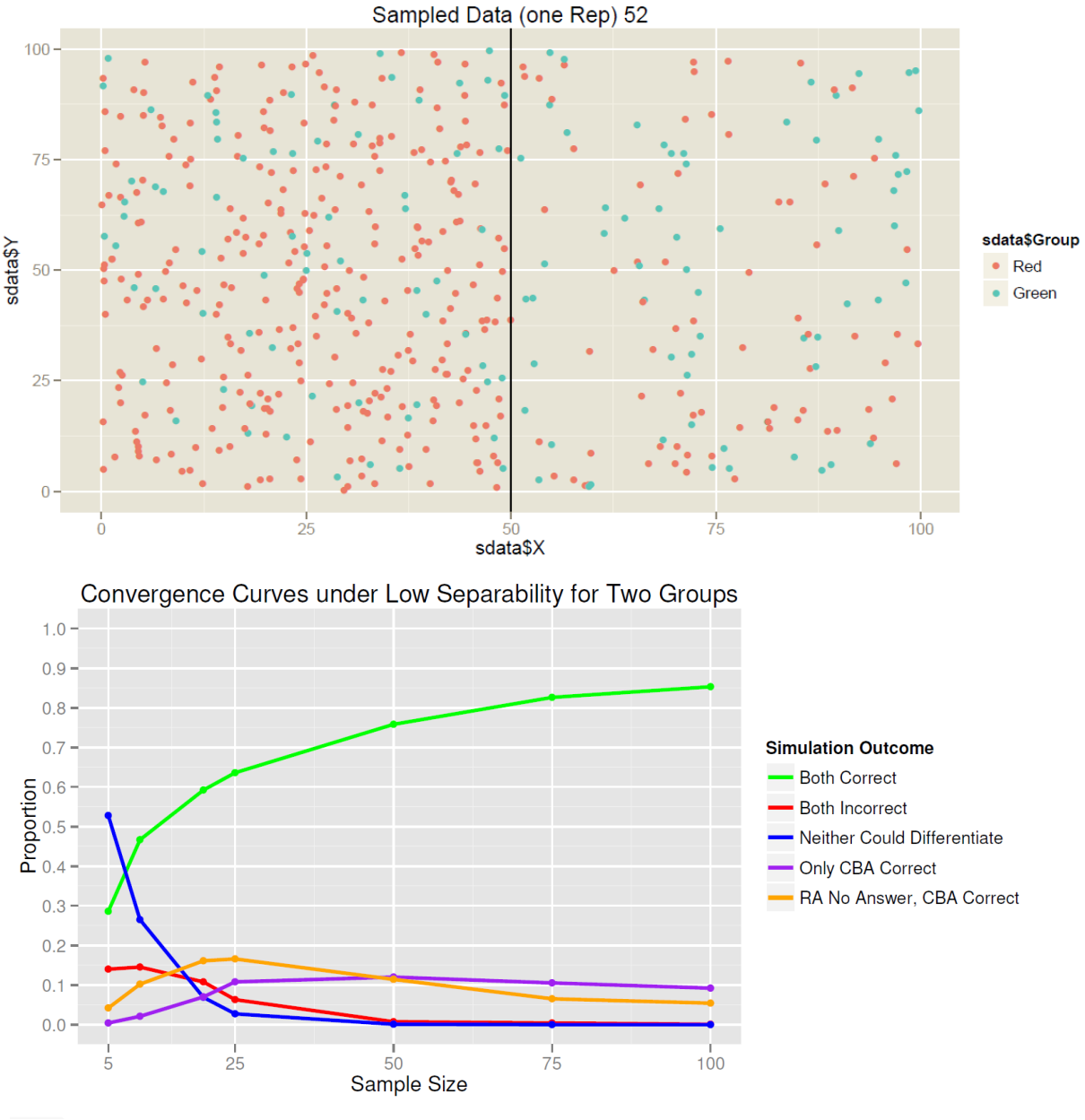

Figure 3.6 Data snapshot and convergence curves for two groups in a scenario with low separability.

measures are able to select the correct model more often, however the rate of convergence is much slower and for those lower sample sizes neither measure could differentiate between the models very often. For these low amounts of data, we see the effect of class imbalance taking shape as neither could differentiate or both made the wrong model choice more often than either model got the correct answer. As we slowly increase the amount of data, class balance accuracy has an advantage over overall accuracy because of its ability to discriminate between classes. It is in these circumstances where class balance accuracy chooses the correct model when regular accuracy cannot differentiate between them or simply selects the wrong model. The benefits of using class balance accuracy do begin to taper off after about 50 observations yet, we see that there is a clear benefit to using class of accuracy over regular accuracy for sample sizes in this range. Putting all of the information together, we see the class balance accuracy is the preferred measure in circumstances of low separability between classes and when there is a small amount of data. This is a telling result because we have shown instances where the use of class balance accuracy will select the model that not only has the highest level of predictive accuracy, but also them be the one that is more descriptive as well.

3.4.3 Study 3: U.C.I. Hold-Out Study

With the results, of our other two study in hand, we now revisit our repository data sets. Given what we know about the performance of class balance accuracy for various levels of concept complexity and amounts of data a final, albeit smaller scale, investigatory study was conducted. Similar to the first, for each data set each, all available models will be fit and our seven measures calculated. Taking a cue from the previous study, holdout samples of various sizes were taken and used as the training sets. These holdout samples start with very little data, using only 25% of the original observations. The model were built from these training sets and then applied to the remaining test observations. This process was repeated five times for each holdout sample, after which, the amount of data for the holdout sample was then increased to 66% and 75%. This hybrid study both emulates the machine learning process as currently practiced by using holdout sets and resampling procedures, but adds a small amount of rigor by varying the size of the holdout samples. This process was applied to all of the data

sets and relevant results such as the mode of the number of groups predicted and the average level of overall accuracy. Less important statistics, such as the average difference in training bias between the training and test sets were kept along with a rounded estimate of the average counts. Though not exactly a full factorial randomized design, this experiment construction will allow us to assess how well the models chosen by class balance accuracy perform on unforeseen test sets data, but also how its selection changes as more data is received.

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.25	cba	forest	5	0.919	0.002	549.8
		fscore	forest	5	0.919	0.002	549.8
		gmean	bayes	5	0.397	0.046	237.2
		ba	bayes	5	0.397	0.046	237.2
		mcc	forest	5	0.919	0.002	549.8
		cen	bayes	5	0.397	0.046	237.2
		oa	forest	5	0.919	0.002	549.8

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.66	cba	forest	5	0.942	-0.011	255.2
		fscore	forest	5	0.942	-0.011	255.2
		gmean	forest	5	0.942	-0.011	255.2
		ba	forest	5	0.942	-0.011	255.2
		mcc	forest	5	0.942	-0.011	255.2
		cen	bayes	5	0.398	0.005	107.8
		oa	forest	5	0.942	-0.011	255.2

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.75	cba	forest	5	0.946	-0.01	188.2
		fscore	forest	5	0.946	-0.01	188.2
		gmean	forest	5	0.946	-0.01	188.2
		ba	forest	5	0.946	-0.01	188.2
		mcc	forest	5	0.946	-0.01	188.2
		cen	bayes	5	0.391	0.001	77.8
		oa	forest	5	0.946	-0.01	188.2

Table 3.21 Hold out study results for the Anneal data set.

Before our discussion, let us reiterate that the selection of models was done after all repetitions were completed. The implication is we are selecting the model that maximizes the

average value of that measure and then comparing these models according to their average test accuracy. Though listed in the proper order, it is more intuitive to start analyzing the results using the largest holdout sample first. By taking backward steps, we see which models performed the best, on average, given the most data and as we step down, gain insight as to if the measures consistently select this model given fewer and fewer observations. This is similar to study two accept we explicitly see which model was chosen for each measure for the various amounts of data.

We began with the annealing data set, whose outcome was mundane, yet will serve as a simple example of how to interpret the results of this study. After creating five randomly sampled holdout sets containing 75% of the data, models were fitted to each data set, the measures were calculated, and averaged across the iterations. When maximizing the average value, six out of the seven measures selected random for as its preferred model. Confusion entropy selected the naive Bayes classifier which unlike the other model performed very poorly. When given only 66% of the data, all measures return the same results. In the situation where measures have to select from models that were fitted with only 25% of the original data in the average training there was more diversity amongst the model selected. Here the G mean and Balance Accuracy both choose the underperforming Bayes model. Therefore when thinking about the results in the correct order, if given very little data the G mean and balance accuracy measures would have selected an underperforming model, and would require more data in order to select a better one.

Results for hepatitis data set were much more interesting. In this instance CBA and G-Mean rightly selected the highest performing model, linear discriminant analysis, for every holdout sample size. The F-Score, a main competitor, selected the second highest performing model support vector machines when given more data despite initially selecting what would later become the highest performing model. These results are interesting because when given a small amount of data most measures selected LDA, but as more data was introduced, the different characterizations of the datum by the various lead them to choose the model that ultimately would achieve the best overall test results. This was true for regular accuracy and Matthew's correlation coefficient whom both historically performed well at selecting models

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.25	cba	lda	2 of 2	0.783	0.21	65.8
		fscore	lda	2 of 2	0.783	0.21	65.8
		gmean	lda	2 of 2	0.783	0.21	65.8
		ba	lda	2 of 2	0.783	0.21	65.8
		mcc	lda	2 of 2	0.783	0.21	65.8
		cen	forest	2 of 2	0.848	-0.077	71.2
		oa	lda	2 of 2	0.783	0.21	65.8

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.66	cba	lda	2 of 2	0.816	0.106	31
		fscore	svm	2 of 2	0.842	0.096	32
		gmean	lda	2 of 2	0.816	0.106	31
		ba	lda	2 of 2	0.816	0.106	31
		mcc	svm	2 of 2	0.842	0.096	32
		cen	bayes	2 of 2	0.647	0.039	24.6
		oa	svm	2 of 2	0.842	0.096	32

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.75	cba	lda	2 of 2	0.793	0.119	22.2
		fscore	svm	2 of 2	0.779	0.161	21.8
		gmean	lda	2 of 2	0.793	0.119	22.2
		ba	bayes	2 of 2	0.65	0.026	18.2
		mcc	svm	2 of 2	0.779	0.161	21.8
		cen	bayes	2 of 2	0.65	0.026	18.2
		oa	svm	2 of 2	0.779	0.161	21.8

Table 3.22 Hold out study results for the Hepatitis data set.

that attained the highest level of overall accuracy.

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.25	cba	forest	5 of 5	0.966	-0.002	3966.2
		fscore	tree	5 of 5	0.96	0.011	3939
		gmean	forest	5 of 5	0.966	-0.002	3966.2
		ba	tree	5 of 5	0.96	0.011	3939
		mcc	tree	5 of 5	0.96	0.011	3939
		cen	bayes	5 of 5	0.896	0.007	3677.4
		oa	tree	5 of 5	0.96	0.011	3939

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.66	cba	forest	5 of 5	0.97	0.003	1804.6
		fscore	tree	5 of 5	0.965	0.009	1795.8
		gmean	forest	5 of 5	0.97	0.003	1804.6
		ba	forest	5 of 5	0.97	0.003	1804.6
		mcc	tree	5 of 5	0.965	0.009	1795.8
		cen	bayes	5 of 5	0.879	0.002	1634.8
		oa	tree	5 of 5	0.965	0.009	1795.8

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.75	cba	forest	5 of 5	0.973	0	1331
		fscore	forest	5 of 5	0.973	0	1331
		gmean	forest	5 of 5	0.973	0	1331
		ba	forest	5 of 5	0.973	0	1331
		mcc	tree	5 of 5	0.968	0.006	1323.8
		cen	bayes	5 of 5	0.918	-0.006	1255.6
		oa	tree	5 of 5	0.968	0.006	1323.8

Table 3.23 Hold out study results for the Page data set.

The outcome of the experiment on both the page and satellite data yielded nearly identical results. Even when given little data, maximizing class balance accuracy selected the model that what later achieve the highest average overall accuracy. Across all sixteen experiments, CBA chose the model with the highest average overall accuracy on fourteen of the data sets and for the two experiments it did not, the model selected was ranked second. Within the context of our previous study, these results are not too surprising. There is further room to investigate

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.25	cba	forest	6 of 6	0.895	0.004	4318.6
		fscore	svm	6 of 6	0.88	0.022	4244.8
		gmean	forest	6 of 6	0.895	0.004	4318.6
		ba	svm	6 of 6	0.88	0.022	4244.8
		mcc	svm	6 of 6	0.88	0.022	4244.8
		cen	nnet	4 of 6	0.251	0.022	1212.8
		oa	svm	6 of 6	0.88	0.022	4244.8

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.66	cba	forest	6 of 6	0.913	-0.001	1997.8
		fscore	forest	6 of 6	0.913	-0.001	1997.8
		gmean	forest	6 of 6	0.913	-0.001	1997.8
		ba	forest	6 of 6	0.913	-0.001	1997.8
		mcc	forest	6 of 6	0.913	-0.001	1997.8
		cen	nnet	5 of 6	0.402	0.007	878.2
		oa	forest	6 of 6	0.913	-0.001	1997.8

Reps	Takeout	Measure	Model	Groups	$\bar{O.A.}$	$\bar{T.Bias}$	$\bar{Counts}$
5	0.75	cba	forest	6 of 6	0.918	-0.003	1475.8
		fscore	forest	6 of 6	0.918	-0.003	1475.8
		gmean	forest	6 of 6	0.918	-0.003	1475.8
		ba	forest	6 of 6	0.918	-0.003	1475.8
		mcc	forest	6 of 6	0.918	-0.003	1475.8
		cen	nnet	5 of 6	0.446	0.002	717
		oa	forest	6 of 6	0.918	-0.003	1475.8

Table 3.24 Hold out study results for the Satellite data set.

exactly why class balance accuracy is able to select the top performing models quicker. Again, this is likely because of its multi-perspective focus on both precision and recall. By seeking out models that account for these metrics across each class, even when it is seemingly not wise to do so, it may suffer higher error on the initial data. However the payoff is subsequently realized when new data is collected that have minority classes represented within the same bounds as the training data. This forced accounting of the minority classes in the initial phase affords better prediction in the latter.

CHAPTER 4. MULTI-CLASS INSTANCE SELECTION WITH CLASS BALANCE ACCURACY

4.1 Introduction

Many contemporary methods for data analysis rely on what may be called the "Goldilocks principle". When the algorithms are supplied with an insufficient amount of data the predictions may not be robust, yet when faced with a deluge of data the techniques become computationally intractable. It has become obvious that advances in data collection and storage have outpaced the scalability of current data mining tools. This is the current conundrum that big data places on analysts (Rickert, 2011). A two-sided approach for dealing with this issue revolves around increasing the scalability and parallelization of learning techniques and/or utilizing data reduction methods that focus on removing missing, repetitious, or incorrectly coded observations. Instance selection is one such automated technique for the latter of the two approaches which seeks to find subsets of the original data set that, when used to train a model, will result in the same or higher predictive accuracy. Ideally, this best subset of training instances will allow models to be learned quickly and still maintain its robustness. For class imbalance problems, the use of instance selection can have can potentially have off-putting results because the measure used to determine the subset if a subset quality is the overall accuracy of the models trained. Throughout this work we have shown that maximizing accuracy has a tendency to neglect minority classes and its use in instance selection is no different. This fact motivated the use of class balance accuracy as an alternative optimization criteria for the instance selection mathematical program. In the following section will discuss some of the basics of instance selection and how class balance accuracy was embedded to make the technique admissible for class imbalance applications.

4.2 Background

To accomplish the desired goal, a wrapper technique was employed as a way to base the instance selection criteria on an accuracy measurement derived from the classifier's output. Within this framework, the subsets which do not result in higher metric values are ignored. In the instance selection literature, wrapper approaches for training set selection have had the most development and hence the motivation for its utilization (Pedrajas, 2011). In a chapter from his "Integer Programming for Instance Selection" thesis, Walter Bennette motivates, conceptualizes and develops a novel reformulation of the wrapper approach as a mathematical programming problem. This formal recasting of a prominent instance selection method as an integer program not only justifies the use of heuristic search methods that are currently employed for the subset selection procedure, but provides a rigorous framework for which modifications can be built upon. By virtue, this research is hinged on Bennett's formulation and Java implementation of the instance selection procedure.

The binary integer instance selection formulation is as follows:

Define:

a_j is the accuracy value of the j^{th} training data subset.

x_j is a binary choice variable for building the model using the j^{th} training subset.

a_{ij} is a parameter set to 1 if an instance, i is in the j^{th} subset and is 0 if otherwise.

I is the set containing all instance choices.

J is the 2^n set that contains all decision variables.

Integer Program:

$$\max \sum_{j \in J} a_j x_j$$
$$s.t.$$
$$\sum_{j \in J} a_{ij} x_j \leq 1 \quad \forall \quad i \in I$$

$$\sum_{j \in J} x_j \leq 1$$
$$x_j \in \{0, 1\} \quad \forall \quad j \in J$$

The main takeaways from this framework are derived from the fact that to solve this instant selection programming problem, a search must ensue across all possible training data subsets to select the one that maximizes some training metric. The large-scale nature motivates the use of heuristic methods to avoid this exhaustive search. To construct the candidate training subsets, a clever backwards selection scheme that fits naive Bayes classifiers to subsets and in an efficient stepwise manner, determines the inclusion or exclusion of individual instances from the resulting training accuracy of that set (Bennette, 2014).

The accuracy assessment of both subsets and instances would ordinarily utilize some overall accuracy measure. Throughout this body of work, we have shown the natural implications of using overall accuracy metrics; wherefore, underrepresented classes, which have the same weight as all others, may be ignored in cases of low separability between the groups. By embedding class balance accuracy into the objective function of the wrapper reformulation, and into the stepwise selection process, we expect the instance selection procedure to overcome majority class bias. A small simulation study was developed to validate this hypothesis.

4.3 Study 1: Accuracy Comparisons between Class Balance Accuracy and Regular Accuracy Maximized Subsets

To initialize the study, three simulated data sets were created to mimic an increasing level of concept complexity. The class imbalance ratio was first fixed at a 10 to 1 ratio of majority to minority group observations for each data set. A straightforward 2 x 3 factorial design with no replication was used to assess the performance of the instance selection process as we oscillate between maximizing regular accuracy and class balance accuracy for each level of separability. The resulting subsets were then used to learn a naive Bayes classifier and the resulting training regular accuracy and class balance accuracy values were recorded for later utility comparisons. Figures 4.1 and 4.2, visually highlight the results for the non-separable and partially separable experimental runs.

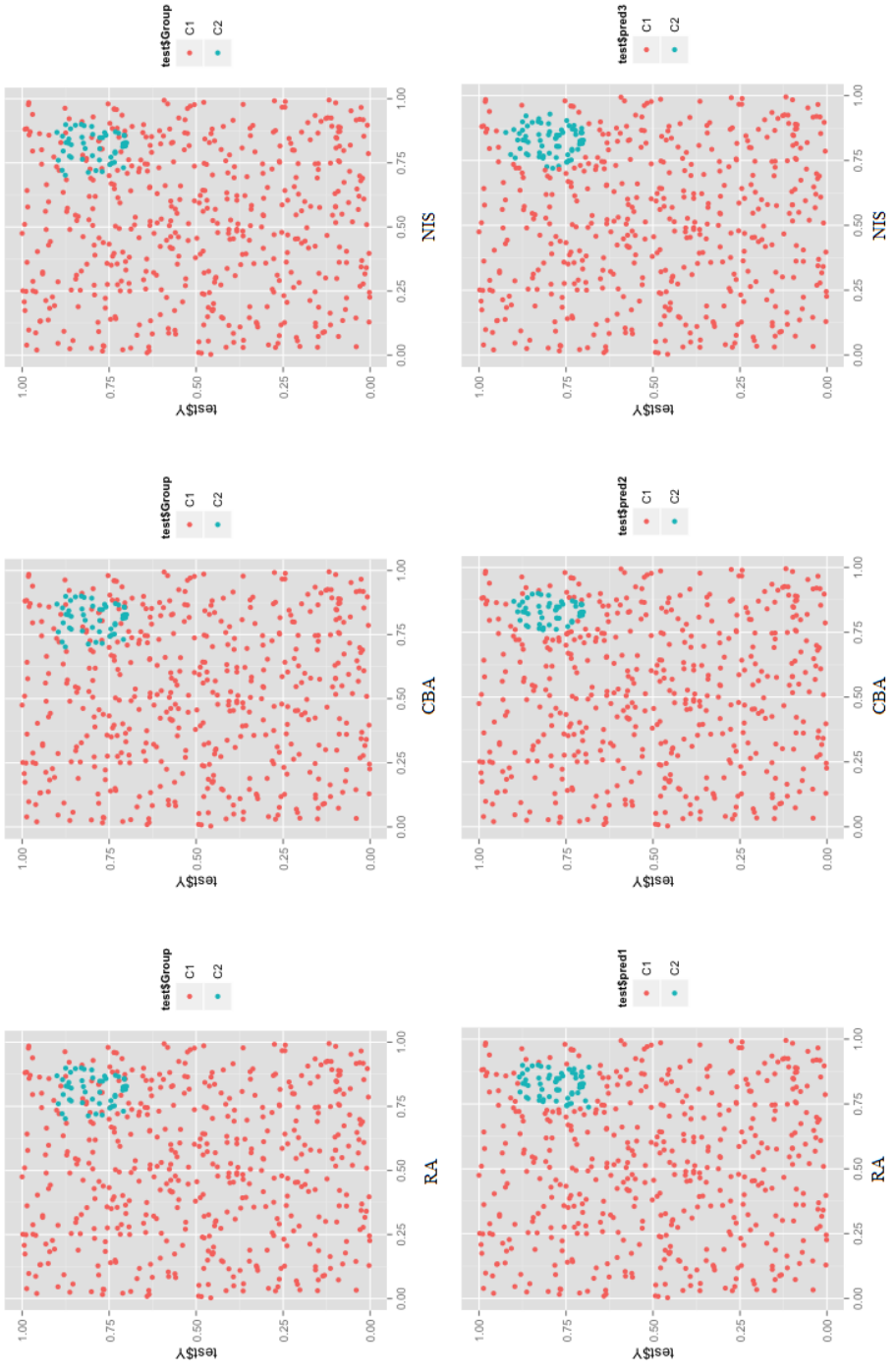

Figure 4.1 Modeling results for Instance Selection derived models after selecting a subset that maximizes Class Balance Accuracy and one that maximizes Overall Accuracy under non-separable concept complexity.

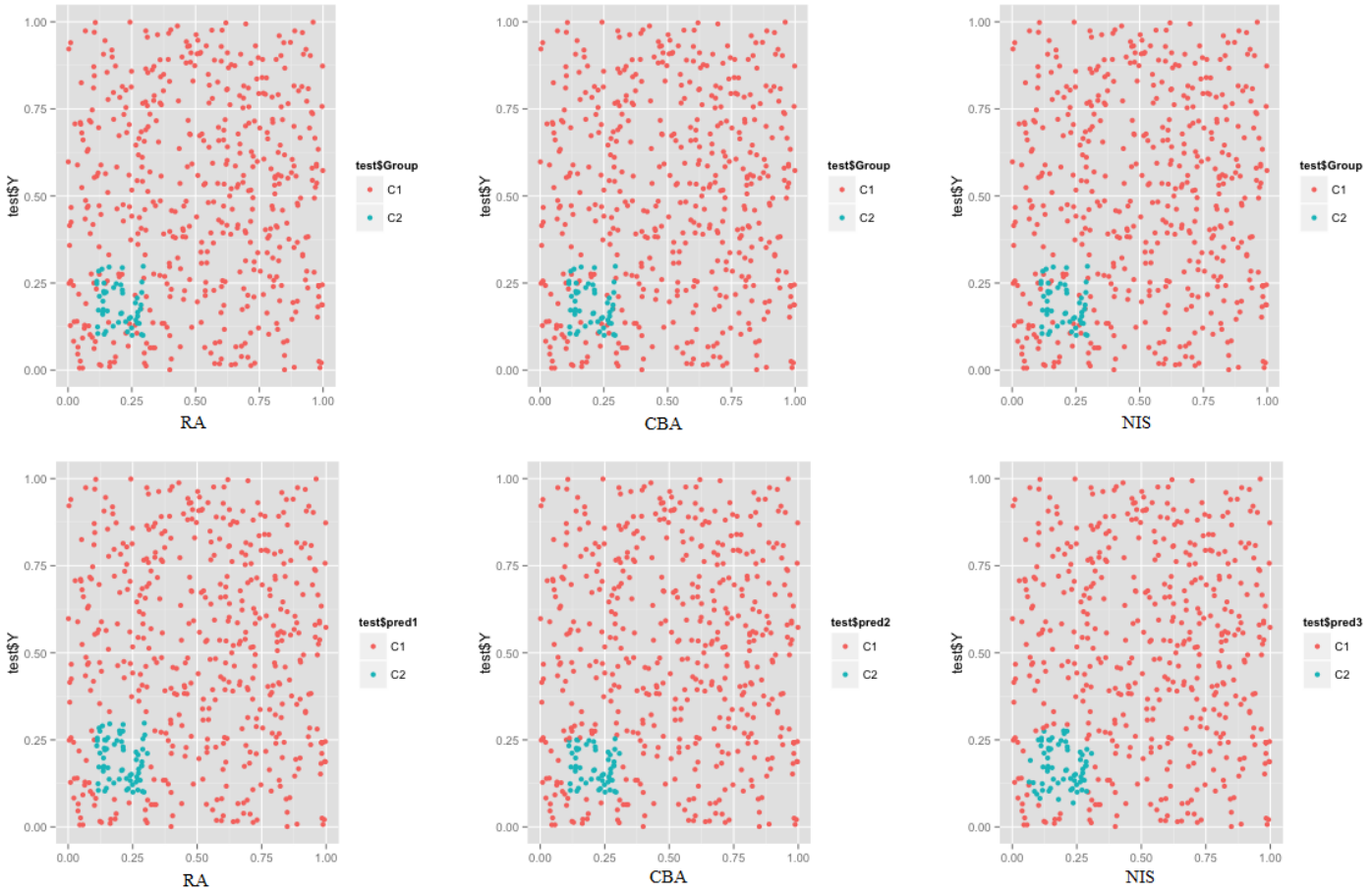

Figure 4.2 Modeling results for Instance Selection derived models after selecting a subset that maximizes Class Balance Accuracy and one that maximizes Overall Accuracy under partially separable concept complexity.

A visual inspection of Figure 4.1 highlights that the classifier derived from the class balance accuracy maximized subset is smaller than its counterparts, fitting seamlessly inside the true area. Minority observations for the non-separable concept complexity data set were uniformly distributed inside of a one unit square block. Along the top row are all identical pictures of the original simulated data set. Underneath each plot is a representation of the predictions for the naive Bayes classifier as applied to the subset maximized by the measure. For robustness, we included a view of the prediction results of fitting a naive Bayes classifier without any instance selection. Overall each classification method appears to recall many of the observations from the minority class in such a way that the results look visually similar. For the partially separated data set, comparable results were achieved. Scrutinizing further into Table 4.1 reinforces a more nuanced understanding of the differences between maximizing these two measures.

Dataset	Maximize	Accuracy	CBA
Non-Separable	RA	0.951	0.825
Non-Separable	CBA	**0.958**	**0.870**
Partially-Separable	RA	**0.971**	**0.863**
Partially-Separable	CBA	**0.971**	**0.897**
Separable	RA	**0.998**	**0.989**
Separable	CBA	0.995	0.978

Table 4.1 Instance selection model results from three simulated data sets. Three degrees of concept complexity were analyzed: Separable, Partially-Separable and Non-Separable. As the concept complexity increases, building models from subsets that maximize Class Balance Accuracy will out preform similar subsets that maximize Regular Accuracy.

The tabular results provide much more resolution into the differences between the results. For the non-separable case, instance selection based on class balance accuracy induced the classifier with better overall and class balance accuracy. Returning back to Figure 4.1, we are reminded that the size of the boundary created by the CBA maximized instances was indeed smaller than its regular accuracy relative. Synthesizing both the accuracy and class balance accuracy training accuracies for this subset hints that the smaller bounds created were more precise, and captured as many of the minority class observations as possible without sacrificing

precision. For the partially separable case we do not receive a gain in overall accuracy, however there is a small positive delta in class balance accuracy. It is interesting that the concept of discriminancy appears again as it becomes apparent that the regular accuracy measure cannot differentiate between the predictive quality of either of the induced classifiers. In the perfectly separable case, we found that regular accuracy maximized subsets performed the best. Within context, these results are not surprising and are consistent with observations made about imbalance measurements for easily separable cases. Therefore it is not surprising that an increase in predictive ability of models trained on preprocessed data sets optimally constructed with instance selection are attained across both measures. Results from the simulation study show that embedding class balance accuracy within the instance selection framework can improve accuracy while accounting for minority class observations, particularly in scenarios without clearly separable bounds.

4.4 Study 2: Accuracy Comparisons between Class Balance Accuracy and Regular Accuracy Maximized Subsets

Our second study was designed to highlight the utility of instance selection for multi-class class imbalance problems as well as to examine the robustness of the previous findings. Here we employ a holdout methodology that includes replication. The first step of the investigatory process involves removing a holdout sample from the original data set and performing two instance selection procedures, maximizing each measure. This resulted in two distinct subsets of data derived from the training sample. A naive Bayes classifier was fit to both instance selected subsets, and then applied to the test set where the final test accuracy for each case was recorded over five repetitions. Due to reduced frictions in data formatting and manipulations, the choice was made to use the diamonds and glass data sets. To increase the computation speed, a subset of the thousand observations from the diamonds dataset were used in lieu of the entire population of over 54,000 data points.

Results from Table 4.2 show the training and test accuracies for both instance selection runs and when no selection scheme was employed. Our focus will be on the test accuracies

for each procedure, as they are the most reflective of model performance. It is immediately reassuring to see that both instance selection procedures outperform the naive Bayes fit to the original data. This reaffirms the overall benefit of instance selection, and touts its ability to remove noisy observations which will ultimately result in subsets that can induce better performing classifiers than those trained on the original sample. When comparing the two instance selection procedures, we see that the subsets chosen by maximizing overall accuracy result in classifiers with higher test accuracy. Likewise, subsets that were derived from maximizing class balance accuracy resulted in classifiers that achieve the highest test values under this metric. These results are consistent across every iteration.

The per class outcomes for the Diamonds data reiterate these results. The increase in class balance accuracy is achieved as the subsets chosen by maximizing CBA shift the focus from solely the "Premium", "Ideal", and "Good" classes towards the "Fair" and "Very Good" groups. Note that the "Fair" class is severely underrepresented in the population accounting for only 3% of the data. When maximizing overall accuracy there appears to be an incentive to ignore this group and therefore the recall values are lower on four the five iterations when comparing the recall across the two selection procedures. For the "Very Good" class, which also happens to be in the minority, the observations benefit from a higher recall when CBA is embedded within the instance selection procedure.

By extracting the first two principal components, a multi-dimensional scaled version of the data was visualized. This two-dimensional representation of the data is plotted along the two independent components which explain the highest proportion of variation from within the variance-covariance matrix, as derived from the data. Figure 4.3 plots the entire data set from which the instances will be derived from. Figure 4.4 shows the actual observations as chosen by the instance selection procedure for both metrics during the first repetition. Though we cannot say definitively how the bounds were drawn, intuitively, we can observe the difference between the two selection procedures, particularly how the classes are represented with respect to their location on the plot making it immediately noticeable that maximizing overall accuracy resulted in fewer observations selected into the optimal set. If the objective is to maximize overall accuracy with the fewest number of observations then the subset chosen by maximizing regular

Iteration	Training Accuracy (NIS)	Test Accuracy (NIS)	Training CBA (NIS)	Test CBA (NIS)
1	0.604	0.527	0.446	0.363
2	0.580	0.557	0.456	0.430
3	0.602	0.527	0.481	0.408
4	0.596	0.590	0.456	0.409
5	0.592	0.566	0.464	0.455
Iteration	Training Accuracy (Max CBA)	Test Accuracy (Max CBA)	Training CBA (Max CBA)	Test CBA (Max CBA)
1	0.673	0.623	0.629	0.503
2	0.628	0.560	0.599	0.468
3	0.680	0.581	0.637	0.504
4	0.688	0.626	0.646	0.509
5	0.667	0.608	0.636	0.545
Iteration	Training Accuracy (Max OA)	Test Accuracy (Max OA)	Training CBA (Max OA)	Test CBA (Max OA)
1	0.676	0.683	0.428	0.417
2	0.712	0.650	0.570	0.477
3	0.701	0.617	0.544	0.475
4	0.692	0.674	0.484	0.471
5	0.718	0.641	0.576	0.497

Table 4.2 Modeling results for the Diamonds data set per repetition by Instance Selection technique.

Test Recall Per Class (Max CBA) Iteration	VeryGood	Premium	Ideal	Good	Fair
1	0.383	0.595	0.874	0.375	0.400
2	0.242	0.696	0.711	0.500	0.625
3	0.329	0.443	0.865	0.323	0.667
4	0.253	0.707	0.858	0.423	0.600
5	0.310	0.614	0.858	0.471	0.538
Test Recall Per Class (Max OA) Iteration	VeryGood	Premium	Ideal	Good	Fair
1	0.198	0.881	0.953	0.531	0.000
2	0.088	0.899	0.930	0.536	0.500
3	0.127	0.772	0.880	0.387	0.500
4	0.184	0.890	0.918	0.423	0.400
5	0.226	0.783	0.925	0.294	0.692

Table 4.3 Per class recall for the Diamonds data set per repetition by Instance Selection technique.

accuracy would naturally be the best choice since CBA focuses on per class performance. The instance selection procedure that maximizes CBA took special care in selecting observations to represent the "Fair" and "Very Good" groups. This is apparent because fair observations, as represented by blue circles, exist at the top and bottom ranges of the second principal component whereas only one observation exists in the regular accuracy maximize subset. "Very Good" observations are represented by golden triangles and are interwoven between the clusters of the different group in the CBA subset. In the case of the overall accuracy maximized set, there are only five representative "Very Good" data points which ultimately result in low class recall.

With regards the use of class balance accuracy, there is a very promising story to be told from the results on the glass data set. On four of the five repetitions, using CBA is the maximizing criteria resulted in subsets that induced classifiers that achieve the highest overall and per class accuracy. Table 4.5, highlights how the inability of overall accuracy to select subsets that accounted for multiple classes eventually resulted in subpar performance across all classes. In many of the iterations, one or more groups were completely left out of the maximized subset which resulted in the inability of the classifiers to predict into any of those classes. Given

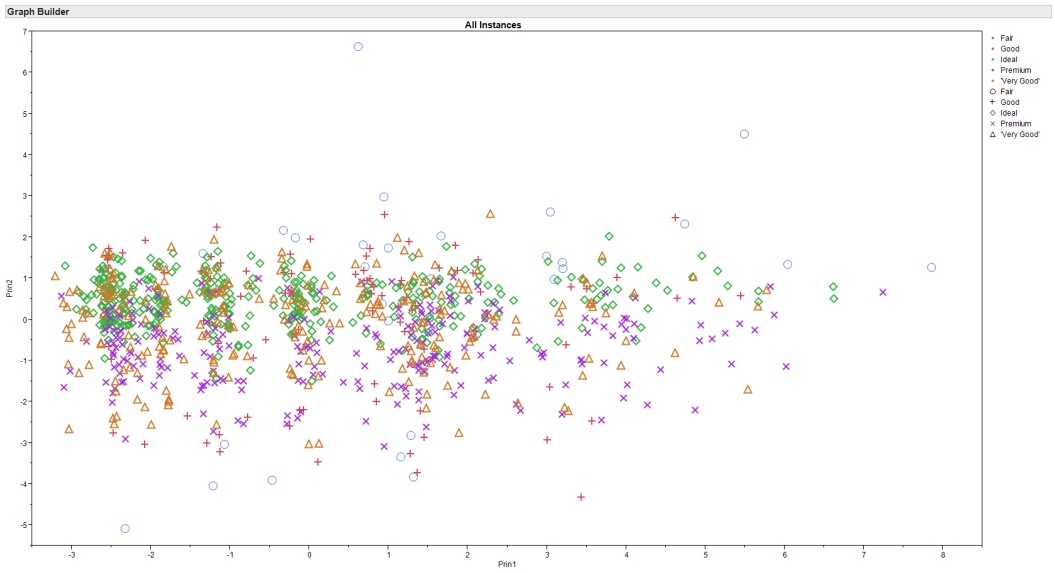

Figure 4.3 A MDS plot of the full Diamonds data set.

the situation with there are multiple classes within a training set, it becomes prudent practice to build classifiers that can account for all classes because of the uncertainty of the group proportions extant in the larger population. This is a crucial result. If the initial model does not account for multiple classes, then when tasked with predicting on unforeseen data, the inability to perceive and demarcate multiple classes can have a potentially devastating effect if the existences of the ignored classes appear in high proportions within the test set. Figures 4.6 shows that for the second repetition group "E", as denoted by golden triangles, was completely omitted from the subset that was maximized by regular accuracy. This resulted in a 100% error rate for this class. As a testament to its ability, the instance selection procedure based on class balance accuracy selected only two observations from the training set to represent the "E" class and was able to achieve 100% recall when the induced model was applied to the test data. By focusing on the per class precision and recall, diverse subsets are selected from the training data which induced robust classifiers that achieved a high level of accuracy overall and across individual classes. These convincing results vet the use of class balance accuracy is an embedded measure within the instance selection framework.

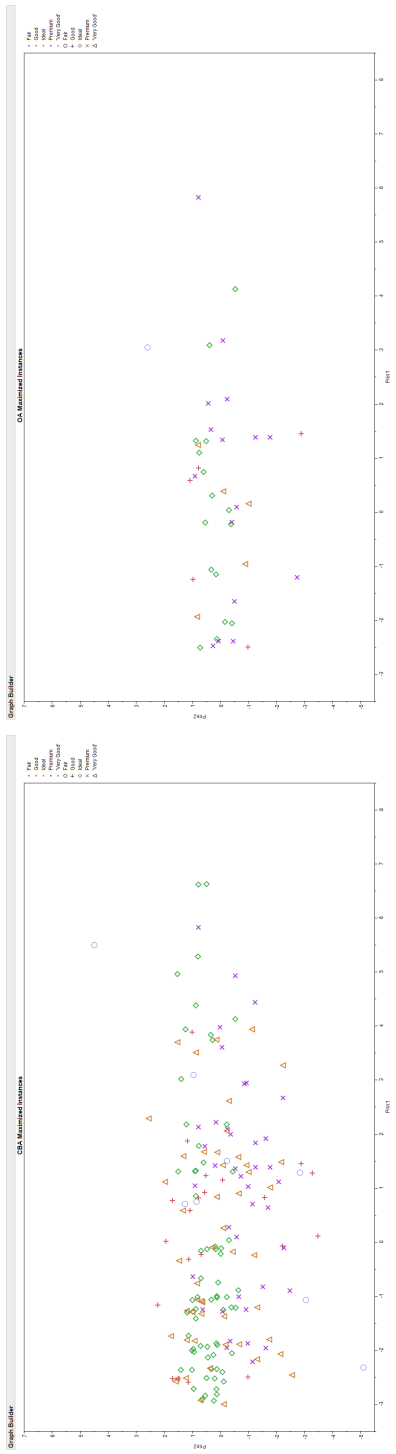

Figure 4.4 Two MDS plots of the instances selected by maximizing CBA (top) and Regular Accuracy (bottom) for iteration 1 on the Diamonds data set.

Iteration	Training Accuracy (NIS)	Test Accuracy (NIS)	Training CBA (NIS)	Test CBA (NIS)
1	0.528	0.417	0.517	0.385
2	0.542	0.375	0.524	0.390
3	0.599	0.486	0.543	0.372
4	0.521	0.444	0.498	0.355
5	0.556	0.472	0.492	0.327

Iteration	Training Accuracy (Max CBA)	Test Accuracy (Max CBA)	Training CBA (Max CBA)	Test CBA (Max CBA)
1	0.768	0.639	0.763	0.518
2	0.690	0.653	0.670	0.514
3	0.746	0.667	0.718	0.546
4	0.725	0.583	0.708	0.478
5	0.789	0.583	0.743	0.432

Iteration	Training Accuracy (Max OA)	Test Accuracy (Max OA)	Training CBA (Max OA)	Test CBA (Max OA)
1	0.768	0.556	0.584	0.318
2	0.768	0.569	0.628	0.330
3	0.768	0.639	0.731	0.548
4	0.683	0.528	0.340	0.273
5	0.775	0.597	0.634	0.355

Table 4.4 Modeling results for the Glass data set per repetition by Instance Selection technique.

Test Recall Per Class (Max CBA)						
Iteration	A	B	C	D	E	F
1	0.667	0.654	0.200	0.400	0.667	0.889
2	0.731	0.630	0.000	0.000	1.000	0.900
3	0.810	0.516	0.500	0.667	1.000	0.818
4	0.640	0.640	0.000	0.250	1.000	0.778
5	0.483	0.783	0.200	0.200	0.667	0.857
Test Recall Per Class (Max OA)						
Iteration	A	B	C	D	E	F
1	0.792	0.462	0.000	0.000	0.333	0.889
2	0.615	0.593	0.000	0.000	0.000	0.900
3	0.619	0.581	0.500	0.333	1.000	0.909
4	0.600	0.680	0.000	0.000	0.000	0.667
5	0.483	0.826	0.200	0.400	0.000	1.000

Table 4.5 Per class recall for the Glass data set per repetition by Instance Selection technique.

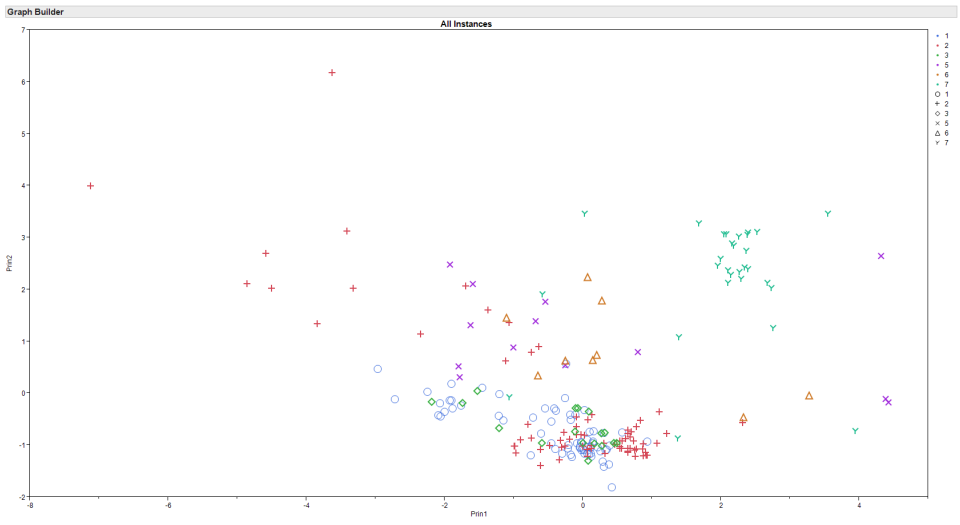

Figure 4.5 A MDS plot of the full Glass data set.

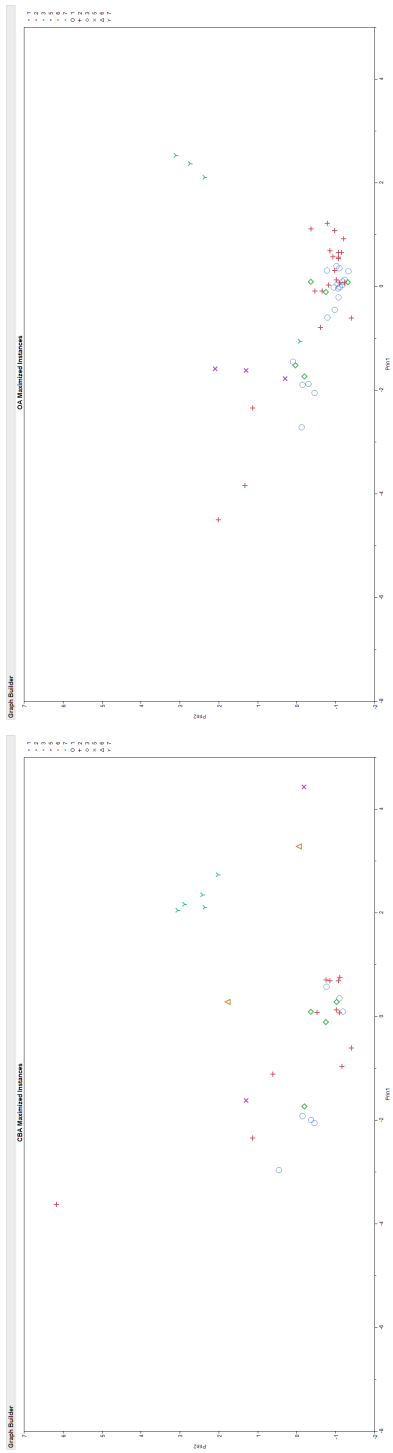

Figure 4.6 Two MDS plots of the instances selected by maximizing CBA (left) and Regular Accuracy (Right) for iteration 2 on the Glass data set.

CHAPTER 5. A NOVEL APPROACH TO MODEL STACKING THROUGH CLASS EXPERT ENSEMBLING

5.1 Introduction

As the field continues to mature, advances in techniques for improving predictions in the presence of class imbalance have been steadily gaining momentum. Survey articles show a diverse number of techniques that revolve around both biased data sampling and algorithm design (Galar et. al, 2012). Galar and others point out that most of the advancements have been in the binary imbalance realm. For these problems the authors show that ensemble methods, which crowd source knowledge from multiple iterative models, can lead to improved class label estimates. In an attempt to apply these methods to the multiclass problem, class decomposition techniques are used to deconstruct the multiclass problem into several binary ones. Wang et al. argue that class decomposition techniques may actually exacerbate the class imbalance problem regardless if "one vs. all" or "one vs. one" methods are implemented. When newer techniques are applied to multi-class problems, it forces a three-step process where the data is first decomposed into some number of sub tasks, and then for each sub-problem an imbalanced technique is applied. At the last step, the class predictions are aggregated across each model fit to every sub-problem. This process is convoluted and has not received much investigation in the literature. As an alternative approach to improving multi-class predictions, we propose the use of Class Expert Ensembling, a novel model stacking technique than leverages model diversity to improve predictive accuracy across each class.

5.2 Background

Class Expert Ensembling is a modeling procedure designed to iteratively partition the data space by having "expert" models make class by class predictions. The full algorithm consists of three main components, expert evaluation and selection followed by a sequential prediction scheme. Stated more formally, given a collection of models, m, and a k-class learning task, we want to select the best preforming model for each particular class. The idea is that some models may outperform other model predictions for certain classes. Typical stacking methods and voting schemes reweight the model predictions for each instance to maximize overall accuracy. CEE seeks to find a given model that specializes in a given class for the learning task. This process is facilitated by the use of a classic integer program called the "assignment problem" where the task is to select among a group of competing units the ones that maximize a specified objective function. For the purposes of Class Expert Ensembling, the solution to this modified assignment problem is a collection of class-model pairs that will be used to make the sequential predictions. Given a new data set, the ordered models are sequentially applied to the data, partitioning and separating each class from the original set as each expert is allowed to access the data. At termination, all observations will be predicted into a class and model assessment can begin.

5.3 Algorithm

The first stage of Class Expert Ensembling involves fitting a collection of models to a given dataset. This collection of models will form the basis of the "multiple classifier system", which will be leveraged to make the predictions. Conceptually, multiple classifier systems are similar to model ensembles where the latter involves techniques that make use of a single model type being perturbed multiple times by some variance inducing procedure, such as resampling, and then aggregating the output predictions. The former induces variation implicitly by allowing models of different types such as Support Vector Machines, Classification Trees, and a host of other algorithms to be admissible within the M.C.S. framework. Model diversity is inherent within the system because of the varied algorithms present and not manufactured through re-

sampling procedures. Before the M.C.S. can be formed, the candidate models are all fit to the given data set and evaluated. As discussed previously, Class Balance Accuracy works as a per class measure that encourages models to improve class Recall while not sacrificing Precision. This property itself fits precisely in the procedural framework despite; in general, evaluating models on a class by class basis is a task that most measures are not well suited. Therefore, though the framework has been designed to accommodate any per class measure, Class Balance Accuracy will be used as one of the main components in the objective function of the binary integer program.

The "Expert Choice Problem", a modified assignment problem, is as follows:

Define:

c_{ij} is a binary choice variable for class i and model j.

e_j is a binary choice variable for model j.

$PCAM_{ij}$ is a Per Class Accuracy Measure for model j's prediction of class i.

OAM_j is an Overall Accuracy Measure for model j.

Where $i = 1,...,k$ classes $j = 1,...,m$ models

Integer Program:

$$\max \sum_{i}^{k} \sum_{j}^{m} PCAM_{ij} c_{ij} + \sum_{j}^{m} OAM_j e_j$$

$$s.t.$$

$$\sum_{j}^{m} c_{ij} = 1 \quad \forall \quad i$$

$$\sum_{j}^{m} e_j = 1$$

$$c_{ij} \in \{0, 1\}$$

$$e_j \in \{0, 1\}$$

The above assignment problem seeks to maximize the sum of the training Class Balance Accuracy contributions across all classes as well as the overall training Accuracy. This objective function represents a mathematical formulation of our desire to select the best model for each class. The solution space, combinations of models and classes, is then constricted by two constraints. The first limits the number of experts per class to a single representative, while the second allows only one overall expert to be chosen.

The formulation of the integer program is motivated by the belief that maximizing the training evaluation criteria will act as a proxy and likewise serve as the combination that will achieve the highest accuracy on any unforeseen test data. This issue is a general data mining problem and not specific to this application, however it must be explicitly stated due to the nature of what is being proposed. Unfortunately, there is no guarantee that another suboptimal combination of experts could not achieve higher accuracy on the test set.

With the optimal model-class pairings, the experts form the multiple classifier system and the foundation is set for the sequential predictions to be made. To complete the Class Expert Ensemble procedure the following 'Assembly Line" algorithm is employed:

Algorithm 1 Assembly Line Algorithm

1: **Solve** the k-Class Expert Ensemble Problem
2: **Select** an Assembly Procedure
3: **if** Procedure = "Class Proportions" **then**
4: **Calculate** Class Proportions from the Training Set, D
5: **Supply** a New Dataset, $D*$
6: **Order** Classes i through k in $D*$ by Ascending Training Set Class Proportionality
7: **else**
8: **if** Procedure = "Per Class Accuracy" **then**
9: **Select** a Per Class Accuracy Measure
10: **Calculate** Per Class Accuracy alues from the Training Set, D

Supply a New Dataset, $D*$
Order classes i through k in $D*$ by Descending Per Class Accuracy
end if
end if
for Each i in k **do**
Make Class Predictions on New Data, D* with Expert j
Remove Predicted Class i Observations from New Data, D*
Next i
end for
Predict D* Remainders with the Overall Expert
end

Making predictions is intuitively simple with this algorithm. Once given a new data set, the experts are first ordered according to the prevalence of the class they intend to predict. Beginning with the class with the least representation, the expert makes prediction on the new data, labeling all observations. Observations that match the model's expertise are removed for the data set and the next model is allowed to make its predictions. It too removed observations within its realm of expertise and steps aside. This process continues across all classes. Because of the nature of the sequencing, there is no guarantee that every observation will be predicted into a class, therefore the overall expert, as denoted by e_i is employed to assign all remaining observations into a class. Once the procedure terminates, all observations will have predicted labels and be ready to assess for statistical accuracy.

5.4 Study: Investigation of Model Performance on Hold-Out Samples from the U.C.I. Model

The experimental design, constructed to compare and contrast the model performance of Class Expert Ensembling, consisted of a simple holdout procedure which used 66% of the

data to train the model and the remaining 33% for prediction and model assessment. Utilizing fourteen data sets, all models were learned on the training set and applied to the hold-out samples. Candidate experts for the multiple classifier system consisted of every model that could be successfully fit to the data, with the exception of adaBoost. This allows for the direct comparison between our expert approach and adaBoost, both of which employ multiple models to make their predictions. For any given data set, every singular model was fit twice, separately and within the Class Expert Ensemble framework.

Table 5.1 contains a ranking of the models according to their overall test accuracy. Along each column, the rank order of each successful model's fit is given for the learning exercise as executed on the data set labeled for that column. At a high level, class expert ensembling as a framework performs relatively well with respect to its peers. Though one single variation does not consistently stand above the crowd, looking down at the results by data set, for almost every data set some variant of class expert ensembling performed well. Four data sets; Annealing, Hepatitis, Balance Scale and Diamonds were modeled best by the class expert ensemble framework. This will be investigated further later in the chapter. As mentioned in the review of literature, ensemble methods as a whole tend to outperform other techniques and this research further supports the claims of previous work in the field as random forests, adaBoost, and class expert ensembles, as a group, generally have the lowest rankings across each data sets. Of the three and, random forests does exceptionally well. An interesting fact to note is that for the two data sets that random forest underperformed, class expert ensembles delivered stellar predictions. By construction, because of its low overall accuracy values, the integer program suppresses the random forest predictions in favor of the expertise of better performing models. From these results, we gain an initial understanding of the benefit of the model diversity that C.E.E. exploits during its model stacking procedure.

When ranking the models according to their test class balance accuracy values, we receive a rather counter-intuitive result. Adaboost and random forests, which are not particularly known for their per class modeling ability, performed relatively well across each of the data sets. Intuitively, we would expect the class expert framework to achieve the best results under a

Model	Anneal	Bal.	Ecoli	Flare	Glass	Hep.	Nur.	Opti	Page	Pen.	Sat.	Seg.	Yeast	Dia.
Trees	4	10	12	6	3	1	8	9	3	7	11	10	10	3
SVM	8	2	1	2	3	1	4	1	9	1	2	6	4	3
LDA		6	3		10	8		4	11	5	6		11	10
Naive Bayes	11	5	5	11	12	13	6	7	13	6	12	11	13	12
Random Forests	6	7	10	1	1	1	2	2	2	2	1	2	1	11
Nueral Networks		4	13	8	13	8	3	8	12	13	13	12	8	3
Adaboost	4	9	10	7	1	12	1	3	1	3	4	1	3	
Climer (CBA,CBA,CP)	10	7	2	12	3	1	8	9	7	7	7	9	6	1
Climer (CBA,CBA,DM)	1	10	5	2	3	1	5	9	5	7	10	4	6	3
Climer (CBA,OA,CP)	1	10	5	2	3	1	8	9	7	7	8	3	2	2
Climer (CBA,OA,DM)	1	10	5	10	3	1	8	9	5	7	9	4	4	3
Climer (BA,OA,CP)	7	1	4	9	11	8	7	5	4	4	5	7	9	9
Climer (BA,OA,DM)	8	2	9	5	9	8	8	6	10	7	3	7	12	3

Table 5.1 Model results ranked according to overall performance using Regular Accuracy.

per class objective because of its procedure explicitly focuses on creating low error class-model pairs. The logic that naturally follows is that the per class optimization of results will result in higher class balance accuracy values for the aggregated predictions. Therefore it is curious that the results don't follow this pattern. The likely result is a consequence of the class balance accuracy accounts for the precision of the predictions made. The class expert framework sequential prediction scheme does not sufficiently constrain the expert models as they select observations into their respective groups. As a consequence, minority groups may achieve high recall but suffer low precision. To gain insight into this, an investigation into the effects of the sequential prediction procedure should be conducted.

To conclude our study, individual results of the Annealing, Balance Scale, and Yeast data sets will be analyzed to get a more nuanced understanding of the class expert ensembling technique. For the Annealing dataset, our multiple classifier system ranked above all other models according to overall accuracy, edging out Classification Trees by one observation. The per class recall for both models appear to be identical, but this is due to truncation. What is of particular interest is that when maximizing class balance accuracy for both per class and overall performance, we return with a multiple classifier system that consists of a combination of random forest and tree classifiers. Individually neither model performed exceptionally well, but when introduced into the expert framework their performance was enhanced. This fact gives support for the utility of this expert procedure. Results on the Balance Scale data set expressed a similar concept. The individual models, when learned separately and applied to the data set underperformed, yet when employed as a unit within the class expert framework decreased the total misclassification error. Though the expert ensemble technique does not outperform for both performance perspectives, its predictions do return a modest 2% increase in overall accuracy. For this variant of C.E.E. tested, regular accuracy was chosen for the overall measure and balance accuracy, the recall per class was chosen as the per class measure of performance. The predictions were made in order of class proportionality with the minority class being predicted first. Though extremely rare for this study, class expert ensembles did outperform the other models according to class balance accuracy on two of the data sets. One

Model	Anneal	Bal.	Ecoli	Flare	Glass	Hep.	Nur.	Opti	Page	Pen.	Sat.	Seg.	Yeast	Dia.
Trees	6	10	12	9	9	2	8	8	3	7	12	10	12	7
SVM	8	3	1	3	8	11	4	1	9	1	2	6	2	6
LDA		9	3		7	7		4	11	5	6		6	3
Naive Bayes	11	7	8	11	12	10	7	7	12	6	9	11	13	2
Random Forests	2	6	9	1	2	11	3	2	2	2	1	2	7	1
Nueral Networks		1	13	2	13	13	2	13	13	13	13	12	8	7
Adaboost	1	2	11	6	1	9	1	3	1	3	5	1	11	
Climer (CBA,CBA,CP)	10	7	2	12	3	2	8	8	8	7	8	9	4	4
Climer (CBA,CBA,DM)	3	10	4	3	3	2	5	8	6	7	10	4	5	7
Climer (CBA,OA,CP)	3	10	6	3	3	2	8	8	7	7	7	3	1	4
Climer (CBA,OA,DM)	3	10	5	10	3	2	8	8	5	7	11	4	3	7
Climer (BA,OA,CP)	7	3	7	6	11	1	6	5	4	4	4	7	9	7
Climer (BA,OA,DM)	8	3	10	8	10	7	8	6	10	7	3	7	10	7

Table 5.2 Model results ranked according to per class performance using Class Balance Accuracy.

of these sets was the yeast sample from the U.C.I. machine repository. Overall performance for the best expert ensemble was found by maximizing class balance accuracy per class, and regular accuracy overall in conjunction with a class proportional prediction sequence. This resulted in 316 correctly classified observations, four shy of the highest ranking model, Random Forest. The gains in class balance accuracy come from the successful prediction of the "ERL" minority class. With so many classes extant in the data, the assignment problem was tasked with finding ten class-model pairs and one overall expert. Across these elecen experts, four distinct models of the original six were chosen. In light of this, intuition suggests that model diversity benefits the modeling process helping to achieve higher overall accuracy.

With the following study we have shown that the novel model stacking procedure that Class Expert Ensembles employs can lead to better overall predictions. Made possible by the use of the per class and overall performance perspectives, class expert ensembles are able to find class-model pairs that additively outperform the singular model learning techniques. Given the imbalanced data sets tested, the use of Class Expert Ensembling and as an algorithmic technique to improve predictions looks to potentially be a promising state-of-the-art method. For further investigation, the expert ensembling framework could benefit from additional investigation into the effects of its class composition scheme and sequential ordering procedures, which have been shown to have some influence on the predicted outcomes.

Model	CBA	OA	Counts
Trees	0.46	0.87	236.00
SVM	0.37	0.84	228.00
Naive Bayes	0.15	0.14	39.00
Random Forests	0.55	0.87	235.00
Adaboost	0.64	0.87	236.00
Climer (CBA,CBA,CP)	0.30	0.55	148.00
Climer (CBA,CBA,DM)	0.46	0.88	237.00
Climer (CBA,OA,CP)	0.46	0.88	237.00
Climer (CBA,OA,DM)	0.46	0.88	237.00
Climer (BA,OA,CP)	0.39	0.85	230.00
Climer (BA,OA,DM)	0.37	0.84	228.00

Table 5.3 Modeling results for the Annealing data set.

Model	A	B	C	D	U
Trees	0.00	0.26	0.99	1.00	0.18
SVM	0.00	0.00	1.00	1.00	0.00
Naive Bayes	1.00	0.77	0.02	0.42	0.09
Random Forests	1.00	0.23	0.99	1.00	0.18
Adaboost	1.00	0.35	0.96	1.00	0.46
Climer (CBA,CBA,CP)	0.00	0.77	0.50	1.00	0.09
Climer (CBA,CBA,DM)	0.00	0.26	0.99	1.00	0.18
Climer (CBA,OA,CP)	0.00	0.26	0.99	1.00	0.18
Climer (CBA,OA,DM)	0.00	0.26	0.99	1.00	0.18
Climer (BA,OA,CP)	0.00	0.26	0.97	1.00	0.09
Climer (BA,OA,DM)	0.00	0.00	1.00	1.00	0.00

Table 5.4 Per class recall for the Annealing data set.

Classes	Experts
A	forest
B	forest
C	forest
D	tree
U	tree
Overall Expert	forest

Table 5.5 Class Expert choices for climer(CBA,CBA,DM) call on the Annealing data set.

Model	CBA	OA	Counts
Trees	0.52	0.77	164.00
SVM	0.60	0.90	191.00
LDA	0.56	0.86	182.00
Naive Bayes	0.59	0.89	189.00
Random Forests	0.57	0.85	181.00
Nueral Networks	0.75	0.90	191.00
Adaboost	0.61	0.87	184.00
Climer (CBA,CBA,CP)	0.52	0.77	164.00
Climer (CBA,CBA,DM)	0.52	0.77	164.00
Climer (CBA,OA,CP)	0.52	0.77	164.00
Climer (CBA,OA,DM)	0.52	0.77	164.00
Climer (BA,OA,CP)	0.68	0.92	194.00
Climer (BA,OA,DM)	0.55	0.82	175.00

Table 5.6 Modeling results for the Balance Scale data set.

Model	B	L	R
Trees	0.00	0.87	0.84
SVM	0.00	1.00	0.99
LDA	0.00	0.99	0.91
Naive Bayes	0.00	0.97	1.00
Random Forests	0.00	0.95	0.94
Nueral Networks	0.40	0.97	0.94
Adaboost	0.05	0.96	0.95
Climer (CBA,CBA,CP)	0.00	0.87	0.84
Climer (CBA,CBA,DM)	0.00	0.87	0.84
Climer (CBA,OA,CP)	0.00	0.87	0.84
Climer (CBA,OA,DM)	0.00	0.87	0.84
Climer (BA,OA,CP)	0.20	0.99	0.99
Climer (BA,OA,DM)	0.05	0.98	0.84

Table 5.7 Per class recall for the Balance Scale data set.

Classes	Experts
B	nnet
L	svm
R	svm
Overall Expert	nnet

Table 5.8 Class Expert choices for climer(BA,OA,CP) call on the Balance Scale data set.

Model	CBA	OA	Counts
Trees	0.373	0.605	305
SVM	0.535	0.621	313
LDA	0.418	0.603	304
Naive Bayes	0.289	0.349	176
Random Forests	0.414	0.635	320
Nueral Networks	0.413	0.615	310
Adaboost	0.389	0.625	315
Climer (CBA,CBA,CP)	0.53	0.619	312
Climer (CBA,CBA,DM)	0.438	0.619	312
Climer (CBA,OA,CP)	0.54	0.627	316
Climer (CBA,OA,DM)	0.532	0.621	313
Climer (BA,OA,CP)	0.403	0.607	306
Climer (BA,OA,DM)	0.402	0.585	295

Table 5.9 Modeling results for the Yeast data set.

Model	CYT	ER	EXC	ME1	ME2	ME3	MIT	NUC	P	V
Trees	0.656	0	0.545	0.684	0.444	0.919	0.57	0.493	0	0
SVM	0.65	1	0.636	0.737	0.444	0.774	0.646	0.557	0	0
LDA	0.718	1	0.636	0.632	0.556	0.774	0.57	0.457	0	0
Naive Bayes	0.012	1	0.727	0.684	0	0.887	0.873	0.2	0	0
Random Forests	0.663	0	0.636	0.737	0.333	0.887	0.582	0.6	0	0
Nueral Networks	0.601	0	0.727	0.684	0.389	0.839	0.658	0.571	0	0
Adaboost	0.675	0	0.545	0.789	0.389	0.919	0.595	0.521	0	0
CEE (CBA,CBA,CP)	0.644	1	0.636	0.737	0.444	0.774	0.646	0.557	0	0
CEE (CBA,CBA,DM)	0.638	1	0.636	0.632	0.5	0.806	0.633	0.564	0	0
CEE (CBA,OA,CP)	0.644	1	0.636	0.737	0.444	0.774	0.658	0.579	0	0
CEE (CBA,OA,DM)	0.65	1	0.636	0.737	0.389	0.774	0.658	0.557	0	0
CEE (BA,OA,CP)	0.632	1	0.727	0.789	0.222	0.871	0.595	0.529	0	0
CEE (BA,OA,DM)	0.687	1	0.545	0.789	0.278	0.871	0.633	0.371	0	0

Table 5.10 Per class recall for the Yeast data set.

Classes	Experts
CYT	tree
ERL	svm
EXC	svm
ME1	svm
ME2	svm
ME3	svm
MIT	nnet
NUC	nnet
POX	bayes
VAC	bayes
Overall Expert	svm

Table 5.11 Class Expert choices for climer(CBA,OA,CP) call on the Yeast data set.

CHAPTER 6. TACKLING CLASS IMBALANCE WITH THE CLIMBR PACKAGE IN R

6.1 Introduction

As interest in the class imbalance problem increases, so has the market demand for software tools that directly address the non-trivial effects they have on prediction and classification tasks. Current solutions to class imbalance issues, such as model evaluation and concept complexity, involve the use of alternative measures, biased sampling, algorithmic modifications and/or a combination of each. With so many avenues of approach, techniques and their implementations are scattered across the landscape of scholarly literature, specifically in statistics, computer science, electrical engineering, industrial engineering, or any field that relies heavily on the analysis of data. Those who search diligently will sporadically find implementations of various approaches, unfortunately, a single repository for class imbalance specific techniques does not exist, forcing practitioners to rely on ad-hoc web searches for techniques and perform code implementations at their own time-expense. It is the author's desire to contribute to the class imbalance body of work by creating a well packaged suite that specifically address the effects of model evaluation and prediction in the presence of skewed distributions.

Following along the footsteps of Frank Harrell's "HMisc" package, the "Class Imbalance in R" package, aptly named "climbR" seeks to be a collection of functions and programming routines that will assist scholars in their supervised learning pursuits. The climbR package seeks to aid in not only the high level conceptual approaches, but the low-level programming nuances that may occur. Again, it is the author's hope that a centralized location for procedures applicable to the class imbalance problem will not only assist those interested in solving one-off tasks, but further spur interest in the field, motivating the creation, publication, and sharing of new

state-of-the-art methods.

Specifically for this body of work, this the climbR package serves as documentation for the implementation of the class balance accuracy measure and the class expert ensemble algorithm. The remainder of this chapter will be dedicated to a walk-through of the current version of the climbR package, focusing on its use in practice. For our exploration, we will be utilizing the balance scale data set from the UCI machine learning repository. Collected from the psychology literature, this data set was originally created to model psychological experimental results, but has useful properties in both dimensionality and size that we will leverage.

```
> str(balance)

'data.frame':        625 obs. of  5 variables:
  class         : Factor w/ 3 levels "B","L","R": 1 3 3 3 3 3 3 3 3 ...
  Left.Weight   : int  1 1 1 1 1 1 1 1 1 ...
  Left.Distance : int  1 1 1 1 1 1 1 1 1 ...
  Right.Weight  : int  1 1 1 1 1 2 2 2 2 ...
  Right.Distance: int  1 2 3 4 5 1 2 3 4 5 ...

> head(balance)

  class Left.Weight Left.Distance Right.Weight Right.Distance
1     B           1             1            1              1
2     R           1             1            1              2
3     R           1             1            1              3
4     R           1             1            1              4
5     R           1             1            1              5
6     R           1             1            2              1

> summary(balance)

 class    Left.Weight Left.Distance  Right.Weight Right.Distance
 B: 49   Min.   :1    Min.   :1     Min.   :1     Min.   :1
```

```
L:288   1st Qu.:2    1st Qu.:2    1st Qu.:2    1st Qu.:2
R:288   Median :3    Median :3    Median :3    Median :3
        Mean   :3    Mean   :3    Mean   :3    Mean   :3
        3rd Qu.:4    3rd Qu.:4    3rd Qu.:4    3rd Qu.:4
        Max.   :5    Max.   :5    Max.   :5    Max.   :5
```

The dataset consists of five variables across 625 complete observations. Predictions will be made on the target variable "class" which consists of three factor levels; balanced, left, and right abbreviated as "B", "L" and "R". Modeling this dataset will task algorithms to partition the classes across a four dimensional space derived from the integer value explanatory variables.

```
> print(qplot(class,data=balance,geom="bar",
fill=class,main="Class distributions for the Balance Scale Dataset"))
```

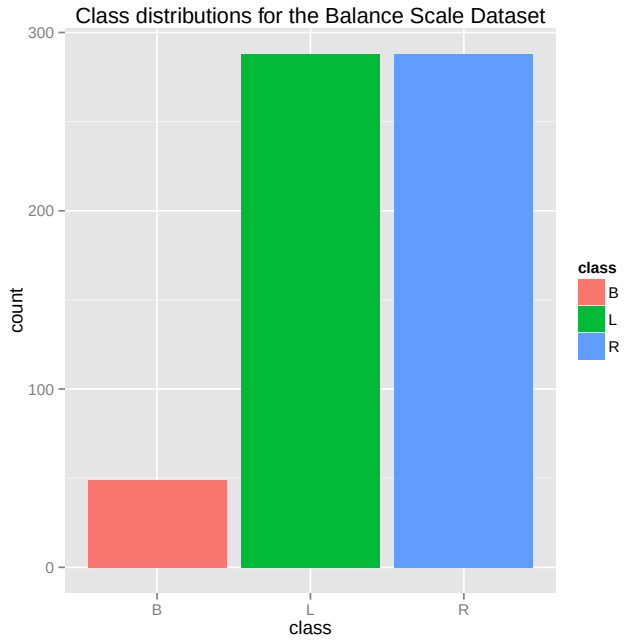

Figure 6.1 Class distributions of the Balance Scale Data.

It becomes apparent that this data set fails to satisfy the assumption of equal class distributions because of the clear underrepresentation of the "B" category. For this data set there exists a "multiple-majority" skew where two classes are identically represented to a much larger

extent than the other. We will begin our analysis by building multiple models with the data and calculating accuracy metrics on the resulting predictions. This process will make use of the "climm" function which will automate the aforementioned process. Ideally, we would like to have a model that performs well overall without neglecting minority class observations.

6.2 climm: Class Imbalance Models and Measures

A common method used to tackle new supervised learning tasks is the "shotgun" approach where all available models are indiscriminately learned on the data. This involves simply fitting as many models as possible to the data set to determine a rank ordering of the models according to the predictive quality of the output. To account for potential over fitting and produce more accurate estimates of the true misclassification error, a subset of the data, often 66% is used to develop the models and the remaining 33% serve as the holdout test set for which the models are applied. By using a holdout sample, we emulate the process of learning models on a training data set and applying these models to make predictions on a new data set with an unknown structure. The hope is that models do not over fit, forming partitions according to patterns that exist beyond the training set, and will perform reasonably well in the absence of known class memberships. Because the groupings are known in the test set, we can rank the models according to their performance on the test data. Therefore the metrics calculated using the contingency table derived from the predicted observations and the test set's known observations will be used to order the models. To add more rigor to the process, the procedure is repeated for set number of repetitions in a bootstrap fashion, where each metric is then averaged across the repetitions to form an unbiased estimate of its true value. This technique is standard in the machine learning community and forms the core functionality of the class imbalance models and measures function, "climm".

The climm function takes the form:

center climm <- function(formula, data, models, measures, reps = 1, takeOut = 1, ...)

Inputs into the climm function include the prediction formula, a reference to the data frame object, a list of models, a list of measures, the user-specified number of repetitions, and the percentage of observations to take out the original data set for model training. The "..."

informally called “dot dot dot” is the ellipsis feature that allows further arguments to be passed on to the embedded functions that support them. As the primary mode of analysis, a one repetition default has been set, along with a holdout proportion of one hundred percent to allow for the prediction of the entire data set.

```
> balance.climm <- climm(
+ class ~ .,
+ data = balance,
+ models = c("tree", "svm", "lda", "bayes", "forest", "nnet"),
+ measures = c("cba", "fscore", "gmean", "ba", "rci",
+ "mcc", "cen", "oa", "counts", "class.cba", "class.recall",
+ "class.precision", "class.fscore", "class.counts"),
+ reps = 5, takeOut = 0.66)

> balance.climm

Data Sets:  train test
Models Fit:  tree svm lda bayes forest nnet
Number of Observations in each Training Set:  413
Number of Reps:  5
```

Here we call the climm function on the balance data set looking to analyze both the per class and overall accuracy of six models set to their respective defaults, ala without parameter tuning. To train the models, two-thirds of the data will be used and then applied to the remaining test samples such that the preceding calculations done on the test set can be averaged across five repetitions. The initial print output returns the number of data sets, the models fit, the size of the training sets, and the number of repetitions. To store the various statistics model fits and measures, a “climm” list object class was created. This special class object stores the model fits and statistics for each repetition throughout the procedure, which allows for diagnostic checking across each iteration.

```
> str(balance.climm)
```

```
List of 2
 \$ train:List of 6
                :
                :
                :
 \$ test :List of 6
  ..\$ ScalarMean   : num [1:6, 1:9] 0.524 0.609 0.58 0.598 0.587 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..\$ : chr [1:6] "tree" "svm" "lda" "bayes" ...
  .. .. ..\$ : chr [1:9] "cba" "fscore" "gmean" "ba" ...
  ..\$ PerClassMean :List of 6
  .. ..\$ tree  : num [1:5, 1:3] 0 0 0 0 0 ...
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  .. ..\$ svm   : num [1:5, 1:3] 0 0 0 0 0 ...
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  .. ..\$ lda   : num [1:5, 1:3] 0 0 0 0 0 ...
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  .. ..\$ bayes : num [1:5, 1:3] 0 0 0 0 0 ...
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  .. ..\$ forest: num [1:5, 1:3] 0 0 0 0 0 ...
```

```
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  .. ..\$ nnet  : num [1:5, 1:3] 0.404 0.571 0.424 0 8.6 ...
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..\$ : chr [1:5] "class.cba" "class.recall"
  .. .. .. ..\$ : chr [1:3] "B" "L" "R"
  ..\$ scalarRepMeas: num [1:6, 1:9, 1:5] 0.523 0.613 0.588 0.604 0.604 ...
  .. ..- attr(*, "dimnames")=List of 3
  .. .. ..\$ : chr [1:6] "tree" "svm" "lda" "bayes" ...
  .. .. ..\$ : chr [1:9] "cba" "fscore" "gmean" "ba" ...
  .. .. ..\$ : NULL
  ..\$ models       : chr [1:6] "tree" "svm" "lda" "bayes" ...
  ..\$ numObs       : int 212
  ..\$ numReps      : num 5
 - attr(*, "class")= chr "climbR.list"
```

Again, list objects have a hierarchical organizational structure that facilitate expedited querying of desired outputs. For climm objects, this breaks down into two primary branches that contain information on the training and test sets. Within each branch, the various average statistics are stored separately for overall performance measures and their per class counterparts.

```
> round(balance.climm$ test$ ScalarMean,3)
```

```
         cba fscore gmean    ba   rci   mcc   cen    oa counts
tree   0.524  0.550 0.000 0.571 0.302 0.628 0.349 0.796  168.8
svm    0.609  0.630 0.000 0.653 0.635 0.843 0.155 0.911  193.2
lda    0.580  0.603 0.000 0.626 0.487 0.770 0.241 0.873  185.0
bayes  0.598  0.620 0.000 0.644 0.579 0.818 0.186 0.898  190.4
```

```
forest 0.587  0.604 0.000 0.612 0.554 0.738 0.280 0.854  181.0
nnet   0.748  0.785 0.777 0.808 0.679 0.832 0.210 0.903  191.4
```

Modeling the Balance Scale dataset with the climm function produced bootstrap calculations for nine evaluation measures for six models. Basing our objective on maximizing overall performance, support vector machines predictions yielded the highest level of accuracy. This is also consistent across the other overall accuracy measures. However, when we shift our focus towards per class performance, we see that neural networks outperformed support vector machines on each of the measures, and particularly on Class Balance Accuracy. We can now look at the per class breakdown of the two top-performing models to give more insight.

```
> round(balance.climm$ test$ PerClassMean$svm,3)

                B      L      R
class.cba       0  0.928  0.899
class.recall    0  0.974  0.986
class.precision 0  0.928  0.899
class.fscore    0  0.000  0.000
class.counts    0 97.600 95.600
```

```
> round(balance.climm$ test$ PerClassMean$nnet,3)

                    B      L      R
class.cba       0.404  0.923  0.918
class.recall    0.571  0.928  0.926
class.precision 0.424  0.950  0.954
class.fscore    0.000  0.000  0.000
class.counts    8.600 93.000 89.800
```

After deconstructing the results per class, we get a clearer picture of the difficulty support vector machines has at finding representative bounds for the "B" class. Neural networks sacrifices some recall performance for the two majority classes to make significant gains in recall

for the minority group. In the end, neural networks, on average, were able to recall 57% of the observations from the "B" membership group.

In this example, we have used the "climm" function to not only fit models but to evaluate them along different criteria. In practice this function can be used to quickly evaluate models, giving the practitioner insight into the type of model that may be useful for her prediction task. In certain situations, if a standout model is found, the climm procedure can be modified to assess multiple models of the same type but with varying parameters reducing the time necessary to fine tune the final model. In this capacity, the climm function acts as a solid initial step for evaluating multiple methods across different objective criterion.

6.3 climer: Class Imbalance Experts

Recall that the class expert ensembling method is a multiple classifier system that uses a novel class decomposition technique and sequential prediction algorithm to help improve predictions in the presence of class imbalance. Within the climbR package, there is a implementation of this procedure that can be called with the "climer" command. It steps through the expert ensembling procedure only after first requiring the user to specify a per class measure, overall measure, and a prediction ordering scheme. The integer program is then solved for the models that perform best across each class and overall. Observations in the training set are ordered according to the selected procedure and the predictions are made on a per class basis by their respective models. Please refer to the algorithm and it's treatment given in Chapter 5 for further details.

Packaged together with the climm function, the climer command attempts to directly improve on predictions on skewed response variables. Its R implementation takes the following form:

function(formula, data, models, perClassMeas = 'class.cba', overallMeas = 'cba', perClassSort = FALSE, ...)

With a function call similar to its climm cousin, many of the input parameters are the same. To train a classifier system the user supplies a formula, dataset, a list of models, a single per

class measure, an overall measure, and sorting procedure. Since the focus of this research is on per class accuracy the default is set to the original class expert ensemble algorithm which leverages class balance accuracy as both the per class and overall measure.

```
>data <- balance
>formula <- class ~ .
>newdata<-resample.cr(data,.66)
>data.tr<-newdata$train
>data.test<-newdata$test
> model.climer.ba.oa.pro <- climer(class ~ .,
+ data = data.tr,
+ models = c("tree","lda", "svm", "bayes", "forest", "nnet"),
+ perClassMeas = "class.recall",
+ overallMeas = "oa",
+ perClassSort = TRUE)
```

For this example, to highlight the climer's versatility, we have chosen per class recall and overall accuracy as the expert selection criteria. In a similar fashion to our last example, the Balance Scale data set was first partitioned into a training and test set, each containing 66% and 33% of the data, respectively. At the end of the modeling procedure the function returns an object of the "climer" class which contains model fits and statistics for the procedure, which can be accessed using the str() command. We use a polymorphic summary function to outputs relevant modeling diagnostics; such as, the experts chosen per class, the overall expert, the ordering by which the predictions were made, and lastly a confusion matrix based on the training data.

```
> class(model.climer.ba.oa.mea)

[1] "climer"

> summary(model.climer.ba.oa.mea )
```

```
 Formula: class ~ .

 Per Class Experts:
  classes experts
1       B    nnet
2       L    nnet
3       R     svm


 Overall Expert: nnet

 Class Order:
  R   L   B
194 187  32

 Confusion Matrix:

classes  B   L   R
      B 12   0  20
      L  1 184   2
      R  0   0 194
```

From the results, neural networks was the preferred model choice for classes "B" and "L". It was also the overall expert, however support vector machines was the stand out model for predicting the "R" class. By using the descending per class measure ordering procedure, classes were lined up according to the descending recall values. Therefore since the "B" class was the most difficult to predict, it was placed in the last position.

```
> climer3b.pred<-predict.climer(model.climer.ba.oa.mea,data.test)
> head(climer3b.pred)
```

```
class        LW        LD        RW        RD        predictions
R        1        1        2        1        R        perClass
R        1        1        2        4        R        perClass
R        1        1        3        2        R        perClass
R        1        1        3        3        R        perClass
R        1        1        4        2        R        perClass
R        1        1        5        2        R        perClass
```

Making predictions on new data sets requires the use of the predict() command. This statement call is generic and requires only the model climer object along with an identifier for the new data set. Unlike other modeling procedures, the predict statement returns an updated version the original data with the predictions appended to the back. At the end of this new data frame you will find a column that indicates if that prediction was forecasted by a per class expert or the overall. This can be meaningful when attempting to diagnose problems in the modeling procedure.

```
> round(res.all,3)
```

```
                       CBA    OA Counts
Trees                0.522 0.783    166
SVM                  0.604 0.906    192
LDA                  0.589 0.882    187
Naive Bayes          0.613 0.920    195
Random Forests       0.592 0.863    183
Nueral Networks      0.701 0.877    186
Adaboost             0.645 0.854    181
Climer (CBA,CBA,CP) 0.621 0.882    187
Climer (CBA,CBA,DM) 0.522 0.783    166
Climer (CBA,OA,CP)  0.621 0.887    188
```

```
Climer (CBA,OA,DM)  0.597 0.892    189

Climer (BA,OA,CP)   0.589 0.882    187

Climer (BA,OA,DM)   0.687 0.910    193
```

```
> round(res.perclass,3)

                          B     L     R
Trees                 0.000 0.832 0.872

SVM                   0.000 0.980 0.989

LDA                   0.000 0.960 0.957

Naive Bayes           0.000 1.000 1.000

Random Forests        0.000 0.941 0.936

Nueral Networks       0.294 0.960 0.894

Adaboost              0.118 0.911 0.926

Climer (CBA,CBA,CP) 0.059 0.960 0.947

Climer (CBA,CBA,DM) 0.000 0.832 0.872

Climer (CBA,OA,CP)  0.059 0.960 0.957

Climer (CBA,OA,DM)  0.000 0.990 0.947

Climer (BA,OA,CP)   0.000 0.960 0.957

Climer (BA,OA,DM)   0.235 0.950 0.989
```

To conclude, our modeling routine returned the second best ranking overall along both per class and total accuracy. If the ultimate objective is to balance both the overall and per class performance, the C.E.E. model is objectively the best choice. As a guided walk through of the balance scale data set, we have shown the value added of using the climm and climer functions to evaluate models and improve our predictive accuracy.

6.4 Utility Functions

The Class Imbalance in R package also includes utility functions to assist in low-level modeling tasks. For model evaluation, each of the measures is stored as a separate function which may

be called on any defined table. The full set of measures implemented are: Class Balance Accuracy, F-Score, Geometric Mean, Balanced Accuracy, Relative Classifier Information, Mathew's Correlation Coefficient, Confusion Entropy, Regular Accuracy, Counts, per class Class Balance Accuracy, per class Recall, per class Precision, per class F-Score, and per class Counts. A calcMeasures() function is included that takes the implemented measures as an inputted list, along with the contingency table, and returns an ordered list of the calculated measures for that table.

A useful function for automatically dividing data sets into training and test samples is included with the resample.cr() function. After subsetting the data, this command creates a list of two data frames containing the partitioned data set.

By far the most useful utility function is makeTable(), which will normalize a non-square contingency table into a kxk square matrix. This is important because often in multi–class imbalance problems with multiple minority groups, the prediction method, try as it might, will often be unable to predict any observations from said groups. Predictions are inferred by the model, however that level of the factor is empty, so when creating a table with the base tabular function the resulting matrix output will be misaligned preventing functions such as sum() and diag() from operating as desired. A call to the makeTable() function will extend out the matrix creating row and/or columns of all zeros.

```
> table(data.teste$class,nnet.pred)

     nnet.pred
      cp im imL imS imU om
  cp  45  0   0   0   0  5
  im   2 17   6   0   1  0
  imL  0  0   1   0   0  0
  imS  0  0   0   0   0  0
  imU  0  2   7   1   2  2
  om   0  0   1   3   1  1
  omL  1  0   0   0   0  1
```

```
  pp   0  1   1   3   0 10
```

```
> makeTable(data.teste$class,nnet.pred)
```

```
    cp im imL imS imU om omL pp
cp  45  0   0   0   0  5   0  0
im   2 17   6   0   1  0   0  0
imL  0  0   1   0   0  0   0  0
imS  0  0   0   0   0  0   0  0
imU  0  2   7   1   2  2   0  0
om   0  0   1   3   1  1   0  0
omL  1  0   0   0   0  1   0  0
pp   0  1   1   3   0 10   0  0
```

6.5 Package Expansion

There are many future usability extensions that can be made to enhance the climbR package. Some low hanging fruit include the inclusion of the CEE multiple classifier system as a default model into the climm function, support for per class and overall model diagnostic visualizations, and the integration of other performance metrics. Since climbR share similar functionality with the "caret" package by Max Kuhn, techniques suitable for class imbalance implemented in that suite could be ported over to broaden climbR's versatility. It is the authors hope that this package will serve as a small initial step for what will become a larger one toward the advancement of the class imbalance field of study.

CHAPTER 7. CONCLUSION

From the beginning, this body of work was inspired, conceptualized and executed with the intent to help address a more contemporary area of interest in the data mining field. As supervised learning applications have grown in breadth, their use in situations where the target variable is skewed towards one or more classes has become more prevalent, increasing the relevance of the class imbalance problem. Since model evaluation is such an integral component of the supervised learning process, as the procedure that determines if the learned model is sufficiently predictive, our focus has been on the study of measures appropriate for use in this special circumstance. Beyond the study of existing measures, the author offers a new performance metric, Class Balance Accuracy, as a contribution to the class imbalance literature.
This dissertation, through theoretical derivation, exercises in example, designed experiments, novel application and investigative studies show that Class Balance Accuracy is a suitable metric for model measurement in the presence of class imbalance. Results highlight Class Balance Accuracy as a conservative, class independent measure of predictive error whose construction can be recast as a simultaneous lower bound of two measures, the average per class recall and precision. Beyond its theoretical properties and characteristics in practice, its use as an embedded optimization criteria was examined and in the case of instance selection, the integration of class balance accuracy brought gains in both overall accuracy and per class recall on data sets with multiple non-separable classes. Similarly, maximizing per class balance accuracy within an expert ensemble framework boosted predictive performance of the multiple classifier system in three of the UCI repository data sets. These results help establish the versatility of this novel accuracy measure.

As a culmination of the effort devoted to addressing the class imbalance problem, an open-source software implementation of the main results and techniques are being released as

a step towards this research's North Star. It is the author's hope that the Class Imbalance in R package serves as impetus for the collection and sharing of implemented routines dedicated to addressing class imbalance modeling issues. In the end, we anticipate this centralized repository to go beyond increasing efficiency, but encourage the advancement of the class and balance literature through open-source reproducible research.

7.1 Future Extensions

Like all time constrained research, there is room left for further investigation. Within the class imbalance literature as a whole, there is a need for an in-depth systematic survey of the performance of binary class imbalance techniques as extended to the multi-class case. Furthermore, there is a need for a consensus driven framework that researchers can use to compare and contrast results not only for new algorithmic prediction methods, but data sampling techniques as well. Specifically for this research, there is an opportunity for further development of the theoretical properties of class balance accuracy particularly around its asymptotics and boundedness characteristics. More complex simulation studies may be conducted to gather further supporting evidence for the measure and to grant insight into its performance in very specific circumstances. With respect to instance selection, the study can be expanded to account for more complex structures in the original data, while making use of different modeling techniques and maximization criteria. The class expert ensemble framework could be advanced by analyzing the algorithm itself, paying careful attention to its performance comparisons not only against other ensemble techniques but class decomposition methods as well. Lastly, the climbR package can benefit from the inclusion of as many multi-class metrics and models as deemed appropriate for the class imbalance problem. Other low hanging fruit include, expanded plot functionality, support for class decomposition techniques, and updated multi-class data sampling methods.

A. ADDITIONAL THEORY AND R IMPLEMENTATION

Glimmer's S: The Set Theory Forefather of Class Balance Accuracy

This work on Class Balance Accuracy was originally derived from prior research into similarity metrics. One novel such metric that ultimately inspired CBA was Glimmer's S. Initially created as a technique to measure the similarity between two categorical variables, it's formulation eventually morphed into matrix notation where it served to compare the similarity between the set of predicted observations and the original observed data. Its definition is as follows:

Definition *Glimmer's S* Let X and Y be sets such that X_i and Y_i are a countable number of levels which contain observations that exhibit the same i^{th} characteristic. Define Nx_i and Ny_i as the total number of observations within each factor level. Therefore, we can define Glimmer's S as:

$$S = \frac{|X_i \bigcap Y_i|}{max(Nx_i, Ny_i|}$$

The intuition behind the similarity metric was to measure a weighted version of the relative frequency, which would account for the maximum number of times a factor level had occured together across both sets. The cardinality of the intersection between X_i and Y_i is then divided by the larger of the two sets. It becomes obvious that this notion of similarity between sets could easily be extended to concept of distance. Similarity values increase when there is a large number of matches or when the cardinality of the sets converge. The defining characteristic of the measure is realized through the normalizing denominator which penalizes dissimilarity

between the set sizes. Intuitively, the set sizes can be different for host of reasons and they should be accounted for in such a manner that increases the likelihood that the two sets are similar under the condition of comparable cardinality. This size normalization attempts to drown out the effect when two sets have a substantial portion of matches, yet differ greatly in sample size. Intuitively if all the observations in both sets occurred together jointly and at the same cardinality, then the two sets would be identical. It is the culmination of the preceeding logic that separated Glimmer's S from other metrics such as Jaccard's Simmilarity or Dice's coefficient.

Class Balance Accuracy Implementations in R

For reference, the R implementations of Class Balance Accuracy are as follows:

C.B.A. Per Class Contributions

```
class.cba <- function(z) {
# let z be a contingency table let x and y be the variables
xlev <- rownames(z)
xlev
ylev <- colnames(z)
ylev
n <- length(xlev)
n
m <- length(ylev)
m
across <- function(u, v, t) {
if(sum(u) == 0 & sum(v) == 0){ return(0)}
else{
       t/max(sum(u), sum(v))
   }
```

```
 }
 xyacross <- array(NA, c(n, m), dimnames = list(xlev, ylev))
    for (i in 1:n) {
        for (j in 1:m) {
            xyacross[i, j] <- across(z[i, ], z[, j], z[i, j])
    }
 }
 return(diag(xyacross))
}
```

Class Balance Accuracy

```
 cba <- function(z) {
  # let z be a contingency table let x and y be the variables
 xlev <- rownames(z)
 xlev
 ylev <- colnames(z)
 ylev
 n <- length(xlev)
 n
 m <- length(ylev)
 m
 across <- function(u, v, t) {
 if(sum(u) == 0 & sum(v) == 0){ return(0)}
else{
        t/max(sum(u), sum(v))
  }
 }
```

```
 xyacross <- array(NA, c(n, m), dimnames = list(xlev, ylev))
for (i in 1:n) {
        for (j in 1:m) {
            xyacross[i, j] <- across(z[i, ], z[, j], z[i, j])
        }
    }
  return(mean(diag(xyacross)))
  }
```

BIBLIOGRAPHY

[1] E. J. Atkinson and T. M. Therneau. *An introduction to recursive partitioning using the rpart routines. division of biostatistics.* page 61., 1997.

[2] G. Jurman and C. , Furlanello. *A Unifying View For Performance Measures in MultiâĂŞ-Class Prediction.* 2010.

[3] Wei J. Wang S. Yuan, X. and Q. Hu. A novel measure for evaluating classifiers. In *Expert Systems With Applications*, 2010.

[4] Liaw A. and Wiener M. *Classification and regression by randomforest.* R news, 2002.

[5] Garcia V. Mollineda R. A. Sanchez J. S. Alejo, R. and J. M. Sotoca. *The class imbalance problem in pattern classification and learning.* In II Congreso Espanol de Informatica, 2007.

[6] Gamez-Martinez Matias Garcia-Rubio and Noelia Alfaro-Cortes, Esteban. *adabag: Applies multiclass adaboost.m1, adaboost-samme and bagging*, 2012.

[7] I. Arel., *Deep machine learning - a new frontier in artificial intelligence research.* Computational Intelligence Magazine, IEEE. 5(4):13–18, 2010.

[8] Ripley B., Atkinson B., and Therneau T. *rpart: Recursive partitioning*, 2013.

[9] K. Bache and M. Lichman. *UCI Machine Learning repository*, 2013.

[10] Bustince H., Fernandez A., Galar M., Barrenechea, E. and F. Herrera. *A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches.* IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews, 42:463–484, 2012.

[11] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* 2006.

[12] L. Breiman. *Random Forests. Machine Learning,.* 2001.

[13] L. Breiman. *Classification and Regression Trees.* 1984.

[14] C. Cortes and V. N. Vapnik. *Support-vector networks.* page 20., 1995.

[15] Ripley B. D. and Venables W. N. *Modern Applied Statistics with S.* Springer, New York. 2002.

[16] Leisch F. Meyer D. Dimitriadou E., Hornik K. and Weingessel A. *Misc functions of the department of statistics (e1071)*, 2012.

[17] Longadge R. Dongre, S. S. and L. Malik. *Class imbalance problem in data mining: Review.* International Journal of Computer Science and Network, 2:83, 2013.

[18] C. Drummond and R. C. Holte. *Severe Class Imbalance: Why Better Algorithms Aren't the Answer.* 2005.

[19] Galar M., Tartas E., B. Sola, H. B. Fernandez, A. and F. Herrera. *A review on ensembles for the class imbalance problem: Bagging, boosting, and hybrid-based approaches.* IEEE Transactions on Systems, Man, and Cybernetics: Part C, 42:463–484, 2012.

[20] D. Hand. *Measuring classifier performance: A coherent alternative to the area under the roc curve.* Machine Learning. Machine Learning, 77:103-123., 2009.

[21]] Izenman A. J. *Modern Multivariate Statistical Techniques.* 2008.

[22] N. Japkowicz. *The class imbalance problem: Significance and strategies.* Proceedings of the 2000 International Conference on Artificial Intelligence, 2000.

[23] S. B. Kotsiantis. *Supervised machine learning: A review of classification techniques.* Informatica. 31:249–268, 2007.

[24] X. Li. *Application of data mining in scheduling. PhD thesis, Iowa State University.*, 2006.

[25] Y. Lee. *EM-algorithms for learning a finite mixture of univariate survival time distributions from partially specified class values.* PhD thesis, Iowa State University, 2013.

[26] Arun Kumar M.N and H.S. Sheshadri. *On the classification of imbalanced datasets.* International Journal of Computer Application, 44:1-7., 2012.

[27] Pitts W. McCulloch, W. *A logical calculus of ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics, 5:115-133., 1943.

[28] Phung S., Nguyen, G., and A. Bouzerdoum. *Learning Pattern Classification Tasks With Imbalanced Data Sets.* Pattern Recognition. 2009.

[29] Zeno G., Lars S., Nguyen, T. *A new evaluation measure for learning from imbalanced data.* 2011.

[30] Chauvin Y., Brunak S., Baldi P., and Andersen C. Nielsen, H. *Assessing the accuracy of prediction algorithms for classification: An overview.* Bioinformatics Review, 16:412 âĂŞ 424., 2000.

[31] C. Pelaez-Moreno and F. Valverde-Albacete. *Two Information-Theoretic Tools to Assess the Performance of Multi-class Classifiers.* Pattern Recognition Letters. 2010.

[32] D. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies,* 2:37 - 63., 2011.

[33] Heiberger R. and Plate T. *Combine multi-dimensional arrays.* 2011.

[34] I. Rish and T. J. Watson. *An empirical study of the naive Bayes classifier.* 2001.

[35] M. Sokolova and G. Lapalme. *A systematic analysis of performance measures for classification tasks.* In Information Processing and Management, 2009.

[36] R Core Team. *R: A Language and Environment for Statistical Computing.* 2012.

[37] S. Wang and X. Yao. *Multi-class imbalance problems: Analysis and potential solutions.* IEEE Transactions on Systems, Man and Cybernetics, PartB: Cybernetics, 2012.

[38] C. Weng and J. Poon. *A new evaluation measure for imbalanced datasets.* 2006.

[39] H. Wickham. *GGplot2: Elegant graphics for data analysis.* 2009.

[40] K. B. Zandin. *Maynard's Industrial Engineering Handbook*, Fifth Edition. 2001.

[41] X. Zhu. *Semi-supervised learning literature survey.* 2006.

[42] *Data Mining for Imbalanced Data: Improving Classifiers by Selective* Pre-Processing of Examples. International Doctoral School Algorithmic Decision Theory: MCDA, Data Mining and Rough Sets Session, 2008.

[43] N. Garcia-Pedrajas *Evolutionary computation for training set selection.* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(6): 512-523, 2011.

[44] W. Bennette. *Integer Programming for Instance Selection.* PhD thesis, Iowa State University, 2014.

[45] J. Rickert. *Big Data Analysis with Revolution R Enterprise* Revolution Analytics, 2011.

[46] Matthews, B.W. *Comparison of the predicted and observed secondary structure of T4 phagelysozyme.* Biochim. Biophys. Acta, 405, 1975.

[47] Freund. Y., Robert S. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* AT&T Labs, 119-139, 1997.