Assessing Classification Methods For Prediction of Customer Satisfaction – Santander Bank Dataset

FALL'16 ISEN 613 Project

Team 13

Adithya Ajith
Muzzamil Bashir
Nirav Thaker
Sushant Chittoor Ravinder

Executive Summary

Importance of the problem:
Assessing customer satisfaction is the most important aspect of staying competitive in banking industry. Dissatisfied customer are more likely to leave and usually it happens without any prior warning. In order to address this problem, Santander Bank provided an anonymized dataset for identification of customer satisfaction at *kaggle.com*.

The problem faced by the bank is that dissatisfied customers usually leave without prior notice. This makes it a difficult job for the bank to anticipate customer dissatisfaction. Hundreds of parameters that may influence customer satisfaction are given in the dataset and a prediction model is to be implemented in identifying dissatisfied customers.

Statement of objective:
This prediction model will help in identification of the customers whose probability of dissatisfaction increases to the point of concern, so that a proactive service approach can be adopted to avoid loss in business. The dataset obtained is of a higher dimension, containing thousands of data points and hundreds of predictors. It is highly imbalanced in favor of satisfied customers, unsatisfied customers constitute a mere 3.95% of the total data points.

Methods used:

As a first step, cleaning of data is performed after an exploratory analysis. Exploratory analysis evaluates the dataset for missing and values, constant, duplicated and correlated predictors. Over sampling techniques were employed to balance the training dataset, oversampling doesn't suffer from any information loss and is preferred over under sampling technique. Sampling techniques are used to remove major class bias in the modelling process. **Random forest** and **Fselector** packages were employed for feature selection to identify significant variables. Models were trained on balanced dataset obtained by sampling techniques and tested on imbalanced original test data. Models were then evaluated on the basis of AUC (area under curve) in ROC analysis, and the corresponding results were reported. Modelling techniques of **logistic regression, LDA, random forest, artificial neural network, gradient boosted machine** and **xgboost** are applied.

Key results: XGBoost performed the best with 0.8416 10fold Cross Validated AUC, on the imbalanced training data. The h2o.gbm model gave the second best performance with 0.8328 AUC.

Implication of results: The modern and sophisticated ensemble models work very well with the imbalanced dataset to give both better prediction performance and fast computational speeds.

Introduction

Importance of the problem:

The dataset provided has 76020 data points and 369 predictors, the predictors are anonymized as the bank does not want to share these information. It is a binary classification problem, with a two level response variable as 'TARGET', factor 1 and 0 corresponds to dissatisfied and satisfied customers respectively. The dataset is highly skewed with 73012 observations belonging to satisfied customers. The percentage of dissatisfied customers come out at merely 3.95 % of the total dataset. This presents two potential challenges in the dataset i.e. dealing with dimensionality and dealing with

1. Dealing with a higher dimensional dataset, using the dataset with existing technique lead to non-convergence.
2. Data is skewed, prediction of minor class presents a major challenge.

Objectives:

The objective of this study is to explore prediction algorithm to build a prediction model which can accurately predict the test data for both minor and major classes satisfactorily.

Scope of work:

The scope of the work includes an extensive data cleansing, filtration techniques to remove duplicated and correlated variables. It is followed by more advanced wrapper and embedded techniques for feature selection including relief and random forest. Cross validation is then applied to evaluate prediction models, since the dataset is highly skewed accuracy rate will not be our measure of performance. Instead, area under curve of ROC curve is reported, and the model with best AUC is selected as the final model. Prediction models of logistic regression, LDA, random forest, artificial neural network, gradient boosted machine and xgboost are applied on both imbalanced and oversampled training dataset. Area under curve is reported for both set of training data and a comparison is presented at the end of the report on algorithm ability to handle imbalance dataset.

Literature review

In order to keep the report limited to guideline page numbers, a brief summary of the reviewed papers is presented below. The complete reviews are attached and can be viewed in the APPENDIX-II section.

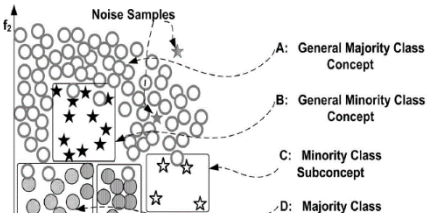
Reference: Haibo He and Edwardo A. Garcia, "**Learning from Imbalanced Data**" IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 9, September 2009
Reviewed by: Adithya Ajith

The challenges for learning from imbalanced data are reviewed in this paper. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms, which assume or expect balanced class distributions or equal misclassification costs. As most of the real world data being generated are imbalanced, the problem of learning from such data has attracted attention from both academia and industry. Due to the unbalanced class distributions, the imbalanced learning problem is concerned with the performance of most standard learning algorithms.

The paper provides a broad survey and review of the development of research in learning from imbalanced data. Also, to develop a comprehensive resource that can be used by data engineering researchers and practitioners. A critical review of the nature of the problem, the solutions or methods, and the current assessment metrics used to evaluate learning performance under the imbalanced learning scenario has been made. Even though unbalanced dataset have inherent issues with the popular learning methods, the methods discussed in this paper will provide a researcher or a practitioner with resources to tackle the problem.

Approach:

By doing an extensive review of the literature available on the topic, the authors have made an attempt to build a toolbox that can be used in machine learning of imbalanced data. The authors first elucidates the nature of the imbalanced data. It is shown how conventional evaluation practice of using assessment criteria, such as the overall accuracy or error rate, does not provide adequate information in the case of imbalanced learning. Different types of imbalances - intrinsic and extrinsic, then relative imbalance and imbalance due to rare instances are explained. Various research suggests that apart from the between class imbalance, data complexity (issues such as overlapping, lack of representative data, small disjuncts, etc) is the primary determining factor of classification deterioration. This data complexity leads to within-class imbalance, as shown in the figure.



A high-complexity data set with both between-class and within-class imbalances, multiple concepts, overlapping, noise, and lack of representative data.

The various available solutions for imbalanced learning are cited and explained in the paper as listed below:

1. Sampling Methods: Sampling technique refers to the different mechanisms that transforms an imbalanced data to have a balanced class distribution. The different methods popular in the research community that are discussed in the paper are : random oversampling and undersampling, informed undersampling, synthetic sampling with data generation, adaptive synthetic sampling, sampling with data cleaning techniques, cluster-based sampling method and integration of sampling and boosting.
2. Cost-Sensitive Methods: As an alternative to sampling methods, cost-sensitive learning targets the imbalanced learning problem by using different penalties that describe the costs for misclassifying any particular data example. The different methods discussed in the paper are: cost-sensitive learning

framework, cost-sensitive dataspace weighting with adaptive boosting, cost-sensitive decision trees and cost-sensitive neural networks

3. Kernel-Based Methods: Kernel methods offer a modular framework for machine learning. First, a dataset is processed into a kernel matrix. Then, a variety of kernel algorithms can be used to analyze the data, using only the information contained in the kernel matrix. The methods discussed includes integration of kernel methods with sampling methods – using SVM and Granular Support Vector Machines, kernel modification methods and active Learning Methods with kernel-based learning methods.

Results

After discussing all the methods used to train a learning model from the imbalanced data, the authors stresses the importance of standardized evaluation metrics to properly assess the effectiveness of each model. The different metrics used across the research community are: singular assessment metrics (precision, recall, F-measure, and G-mean), receiver operating characteristics (ROC) curves, precision-recall (PR) curves and cost curves. S**ince our project dataset is highly imbalanced, this paper provides a framework on how can we better tackle the problem of imbalance.**

Reference: Stephan Dreiseitl and Lucila Ohno-Machado , "**Logistic regression and artificial neural network classification models: a methodology review** " Journal of Biomedical Informatics,Volume 35, Issues 5–6, October 2002, Pages 352–359, Science Direct

Reviewed by: Adithya Ajith

Objective:

The paper is focused on understanding the differences and similarities of Logistic Regression and Artificial Neural Network Classification models, in comparison to other machine learning models. Also to determine factors need to be considered when judging research results using predictive models. Since mistakes in modeling and evaluation can have huge impact on the reported results, care must be taken to ensure that the models are validated and evaluated correctly. The authors make use of available research journals on predictive modelling in the field of medical domains to compare logistic regression and artificial neural network models. Analyze its performance and popularity as compared to the other modelling techniques such as Decision trees, KNN and Support Vector Machines.

By analyzing the model with respect to criteria such as data set size and performance measure, the authors point out which factors need to be considered when judging research results using predictive models. These findings can be deployed in the future predictive modeling research.

Approach:

Comparison of features of SVM, Logistic regression, ANN, KNN and Decision trees is done and its respective advantages and disadvantages listed. The authors point out that SVM is build separating boundaries by solving quadratic optimization problem and since varying degree of nonlinearity and flexibility can be included in the model. The disadvantage of SVM being the classification result is dichotomous, and no probability of class membership is found out. In KNN, the only adjustable parameter being the number of closest neighbors to estimate the class membership, there is not much information on model construction, but they can provide good explanation for the classification result. Decision trees utilize the criterion of information gain while modeling and that they are easily interpretable although this modelling deploys a greedy approach.

Logistic regression and ANN differ from KNN,SVM and Decision trees in the sense that they provide functional forms and parameter details by maximum likelihood estimation, the difference being the first is parametric and

the latter semi parametric /non parametric. In Logistic regression, there is scope for including interaction and non-linear terms making the model flexible at the expense of overfitting. The functional form of ANN is significantly different from Logistic regression, and due to the non-linearity of hidden neurons the model is more flexible than logistic regression. In the Parameter estimation technique, logistic regression being less complex when no non-linear or interaction terms are used reduces the risk of overfitting. To restrict the complexity of neural networks, the author suggests using regularization called weight decay early stopping. The Bayesian framework is believed to be an alternative to solve problem of overfitting. With respect to variable selection, logistic model is more popular due to the simplicity of modelling with forward, backward and step selection. For neural networks, automatic relevance discrimination or sensitivity analysis is used to assess the importance of predictors.

The trained model can be evaluated by using the criteria of discrimination and calibration. Discrimination is a measure of how well the two classes in the data set are separated and calibration determines how accurate the model class probability estimate compared to the true probability. Validation, cross validation or bootstrap is used in evaluating these criteria. Common metrics of discrimination are sensitivity, specificity, accuracy and the area under the ROC curve. In order to calibrate, difference between the average observation and the average outcome of a given group is measured.

The authors further analyzed the 72 papers with respect to the following criteria: whether details of the model building process are given (variable selection scheme for logistic regression, parameter selection and overfitting avoidance for artificial neural networks), whether unbiased estimates of the generalization error are reported (by using test sets, cross-validation, or bootstrapping), whether measures of discriminatory power were given (and statistical testing using these measures), and whether calibration information is included. Also the performance comparison of both the models were made.

Results

The authors arrive at the conclusion that white box models such as Decision trees, Logistic regression offer better interpretability whereas black box models such as SVM and Neural networks have better predictability. As per the findings, the authors conclude that the performance of logistic regression and ANN is superior to KNN and decision trees in most of the experimental cases whereas SVM shows comparable results. The popularity of logistic regression is related to its low generalization error. **Since we will be using both logistic regression and ANN, this paper provides a well evaluated analysis on the benefits of the two methods.**

Reference: A Comparative Assessment of the Performance of Ensemble Learning in Customer Churn Prediction by Hossein Abbasimehr, Mostafa Setak and Muhammad Tarokh, The International Arab Journal of Information Technology, Vol. 11, No. 6, November 2014.

Reviewed by: Muzzamil Bashir

Objective

The objective of this paper is to find the prediction model that gives the best result in classifying customers with a high probability to leave. This classification will help in directing marketing resources towards customers more prone to leaving. This paper uses data analytics to find the best prediction model for customer churn by evaluating performance indicators (AUC, sensitivity and specificity) for different analytics ensemble models. It has been shown that ensemble methods (methods that use a combination of classifier models) shows a better performance over individual models in predicting customer churn. Of the four methods evaluated in the paper, two haven't been evaluated for churn prediction previously.

Approach

The authors address the problem of predicting customer churn by reviewing literature and finding that ensemble methods have shown improvement over base learning techniques. In addition to Boosting and bagging that have been more commonly studied for this type of predictions, authors additionally evaluated stacking and voting techniques.

In the feature selection step, PART (partial decision tree) algorithm was applied to reduce the number of features. The data used in this case was imbalanced due to skewed distribution among the classes. Both basic and advance sampling techniques were applied to reduce the imbalance among the classes. Oversampling and Synthetic minority over-sampling technique (SMOTE) were applied on all four base learner models, resulting in a total of 8 base learner models. Base learner algorithms (C4.5, ANN, SVM, RIPPER) were applied on the balanced dataset, best parameters of each of these algorithms was obtained. Performance criteria for selection of the best prediction model was based on area under curve AUC after application of ROC analysis.

The best base classifiers were then further augmented through use of ensemble techniques of Bagging, Boosting, Staking and voting. All 16 resulting models were evaluated based on the performance criteria and reported.

Results

Table 3. Ensemble learning results.

Ensemble method	Base learner	AUC	Sens	Specs
Bagging	C4.5	0.999	1	0.984
	RIPPER	0.997	0.942	0.98
	ANN	0.941	0.862	0.972
	SVM	0.987	0.987	0.96
Boosting	C4.5	1	1	0.986
	RIPPER	1	1	0.988
	ANN	0.966	0.875	0.984
	SVM	0.966	1	0.962
Staking	C4.5	0.992	0.973	0.962
	RIPPER	0.988	0.978	0.99
	ANN	0.998	0.978	0.991
	SVM	0.978	0.982	0.974
Voting	ALL base learners	0.998	0.978	0.984

Table 2. Base learners results.

Technique	Sampling technique	AUC	Sens	Spec
C4.5	Oversampling	0.983	1	0.958
C4.5	SMOTE	0.886	0.786	0.963
Ripper	Oversampling	97.7	0.977	0.977
Ripper	SMOTE	88.3	0.773	0.985
ANN	Oversampling	0.94	0.87	0.981
ANN	SMOTE	0.92	0.844	0.936
SVM	Oversampling	0.969	0.978	0.96
SVM	SMOTE	0.876	0.835	0.917

It was be inferred that the basic sampling technique of oversampling provides much better results as opposed to the advanced SMOTE. It is also evident that the Boosting ensemble technique based on C4.5 and RIPPER algorithms provide the best result for this case.

In order to improve performance of prediction model, ensemble technique can be applied. The paper investigates two types of sampling & four types of ensemble techniques on the base learning algorithms. It was noted that oversampling improves the prediction performance, and that the application of ensemble significantly improved the predictive power of the base learners. Boosting came out to be the best candidate for churn prediction tasks. **Since our project is related to customer churn, this paper will help in applying the ensemble methods for prediction.**

Reference: An Introduction to Variable and Feature Selection, Isabelle Guyon, Andre Elisseeff. Journal of Machine Learning Research 3 (2003) 1157 - 1182

Objective:

The rapid increase in the domain of data analytics have resulted in problems with a much greater number of predictors, most of which could be irrelevant or redundant. Performing analysis on huge set of predictors is not only computationally intense but also complicates the underlying processes that resulted in the data. The paper discusses the problem of dealing with thousands of variables in the dataset. Selection of variables that would improve the prediction and computational performance of the chosen classification algorithm are extensively discussed. The paper investigates various variable selection techniques and their limitation across a wide domain of study.

The authors are addressing the problem through discussing the available techniques of variable reduction, their limitation and comparative advantages. Various methods of variable selection including ranking, feature construction and subset selection (wrapper, embedded, filter) are discussed in detail.

Approach:

The paper reviews a range of topics on variable selection in data analytics. The study uses datasets from a wide variety of study domains. It starts with discussion of ranking techniques that rank individual variables through correlation coefficients and their limitations through use of constructed examples.

Variable subset selection is then discussed as a solution to the limitations of the ranking techniques. Wrapper (scoring subsets of variable according to their predictive power), embedded methods (variable selection in training) and filters (pre-processing selection of variables independent of chosen predictors) are discussed. Comparative advantages of these three methods are discussed.

Transforming original variables into dimensionally reduced set of features is called feature construction. The paper discusses this aspect of dimensional reduction and its applicability. The paper then discusses approach for cross validation. Non parametric variable selection can be applied by introduction of fake variables in the dataset, these variables can provide elimination criteria for the variables that are comparatively insignificant to the fake variables. The paper concludes with some of the problems that the author encountered during the course of his cross domain study.

Results:

Discussing limitation of ranking techniques (removing individual insignificant variables), the author states through relevant examples that there could be variables that individually are useless but can result in prediction improvement when used with other variables.

Filters are faster than wrappers and can be employed as a preprocessing step to reduce dimensionality and overcome fitting. Wrapper are simple and uses the knowledge from machine learning for variable selection. Wrapper can be computationally intense, however greedy search strategies of forward selection and backward elimination are computationally advantageous and avoid overfitting problems. Embedded methods like decision trees and CART are comparatively faster and have built in capability to variable selection by avoiding splitting of data into training and validation. When using fake variables, a halting criteria can be applied in forward selection.

The paper is an effort to provide a benchmark for variable selection by discussing all the relevant techniques available. It notes that the advantage of using wrapper or embedded methods does exist but it is not universal across different domains. Due to this, the paper is unable to establish a unifying criterion for variable selection

and recommends the use of domain based criterion by evaluating both correlation coefficient and nested subset selection. This paper is highly cited (8563 citation) and provides an in-depth discussion on each of the variable selection methods. **These discussion would be insightful in variable selection for the project.**

Reference: Improved decision making in customer churn prediction context using generalized additive models by Kristof Coussement, Dries Benoit and Dirk Van Den Poel.
http://www.sciencedirect.com/science/article/pii/S0957417409007325

Reviewed by: Nirav Thaker

Objective:

This paper shows that generalized additive models GAM are better over logistic models in prediction of customer ability to leave (customer churn). The authors use a dataset provided by Belgian newspaper publishing company. A comparative evaluation of logistic and GAM model.

Approach:

1) A methodological link between the logistic models and GAM
2) Generalized additive models are represented by replacing a term in the logistic equation with an additive term, as follows:

$$\text{logit}\{P(X)\} \equiv \log\left\{\frac{P(X)}{1-P(X)}\right\} = \alpha + \sum_{j=1}^{p} \beta_j' X_j$$

$$\text{logit}P(X) \equiv \log\left\{\frac{P(X)}{1-P(X)}\right\} = \alpha + \sum_{j=1}^{p} s_j(X_j)$$

3) Feature selection techniques are used: In churn modelling, a subset is selected of the original predictive variables to improve the comprehensibility.
4) The factors used to evaluate the classifiers are a) top-decile lift and b) AUC

The above proposed methodology was then applied to the data is provided by the largest newspaper publishing company in Belgium. The response is 0 and 1 respectively, representing if they renewed the subscription within a 30 day period or not.

Results:

To build the model, Mod_Log selects 18 predictive variables where as Mod_GAM selects 27 predictive variables. After going through the details, it is realized that the predictive terms in GAM are much more significant than they the ones in Logistic. The accuracy of GAM is better than that of Logistic models when tested.

Test set performance in terms of AUC and top-decile lift.

	AUC	Top-decile lift
MOD_LOG	0.8298[a]	4.08
MOD_GAM	0.8452[a]	4.26

[a] Significant with $\chi^2 = 187.40$, $df = 1$, $p < 0.001$.

The **interpretability** of the GAM models is also better. The relaxation of linearity constraint allows the data scientists to have more insight into the explicative variables.

The data shows how the company's profits are affected by using the GAM technique. The marketing analysts are interested in profits as little as 10% when they can predict the customer satisfaction rate batter. The satisfaction rate is directly resulted into company's profits and increased subscription. It is advisable to use GAM techniques to improve the power of customer churn prediction churn model.

This study summarizes the literature on GAM models for improving the accuracy of customer satisfaction prediction models or churn prediction models. We establish that it is beneficial to use the GA techniques to improve on the accuracy and thus improve on the profits.

However, to validate the modeling techniques externally, GAM can be applied to situations other than churn contexts. It may also be applied to classification problems other than binary. They may be related to customer relationship management or other fields. Also, this study only compares the logistic models with GAM, future research can be done where comparison is between models that can both fit non-linear data, like decision trees, neutral networks, random forest, etc. **Since our study deals with customer churn, this paper provide insight to approach such type of problems.**

Reference: Do we need hundreds of classifiers to solve real world classification problems by Manuel Fernandez-Delgado, Senen Barro and Eva Cernadas; http://www.jmlr.org/papers/v15/delgado14a.html

Reviewed by: Nirav Thaker

Objective:

a) Key problem: Identify the best classifiers by testing them on 121 datasets with 179 classifiers.
b) Key reason for the paper: To identify the best performing classifier overall.
c) Key objective: The objective is to achieve conclusions on classifier behavior, not depending on the data at hand. So, we perform the modeling on 121 data sets available on UCI.
d) How did the authors address the problem: They evaluated 179 classifiers arising from 17 families and tested them on 121 data sets. This would give conclusions that could be applicable to any real world problem.
e) What did they find: The models achieving maximum accuracy are RandomForest, SVM and LibSVM in the same order.
f) What does it imply: The difference between the accuracy of RandomForest and SVM is not significant but RF performs better overall.

Approach:

179 classifiers used are implemented in C/C++, R, Matlab, Weka.

179 classifiers with 121 datasets give a combination of 21,659 classifier-dataset. There were errors in many of the classifiers due to various reasons like collinearity, discrete inputs, patterns, etc. These errors are excluded to calculate the average accuracy. Cross validation is performed using the whole available data. Friedman ranking and Cohen K is calculated for each classifier.

However, this methodology may not be applied to large data sets.

In the end the results are those over 4 tests.

Results:

The best ranked classifier is ParRF_t and the second is rf_t. This result is surprising considering that RandomForest is an old method compared to the various new ones and it still performs better. Probability of

achieving the Maximum accuracy is also calculated which shows the number of data sets in which it achieves the maximum accuracy.

No.	Classifier	PAMA	No.	Classifier	PAMA
1	elm_kernel_m	13.2	11	mlp_t	5.0
2	svm_C	10.7	12	pnn_m	5.0
3	parRF_t	9.9	13	dkp_C	5.0
4	C5.0_t	9.1	14	LibSVM_w	5.0
5	adaboost_R	9.1	15	svmPoly_t	5.0
6	rforest_R	8.3	16	treebag_t	5.0
7	nnet_t	6.6	17	RRFglobal_t	5.0
8	svmRadialCost_t	6.6	18	svmlight_C	5.0
9	rf_t	5.8	19	Bagging_RandomForest_w	4.1
10	RRF_t	5.8	20	mda_t	4.1

No.	Classifier	P95	No.	Classifier	P95
1	parRF_t	71.1	11	elm_kernel_m	60.3
2	svm_C	70.2	12	MAB-LibSVM_w	60.3
3	rf_t	68.6	13	RandomForest_w	57.0
4	rforest_R	65.3	14	RRF_t	56.2
5	Bagging-LibSVM_w	63.6	15	pcaNNet_t	55.4
6	svmRadialCost_t	63.6	16	RotationForest_w	54.5
7	svmRadial_t	62.8	17	avNNet_t	53.7
8	svmPoly_t	62.8	18	nnet_t	53.7
9	LibSVM_w	62.0	19	RRFglobal_t	53.7
10	C5.0_t	61.2	20	mlp_t	52.1

No.	Classifier	PMA	No.	Classifier	PMA
1	parRF_t	94.1	11	RandomCommittee_w	91.4
2	rf_t	93.6	12	nnet_t	91.3
3	rforest_R	93.3	13	avNNet_t	91.1
4	C5.0_t	92.5	14	RRFglobal_t	91.0
5	RotationForest_w	92.5	15	knn_R	90.5
6	svm_C	92.3	16	Bagging-LibSVM_w	90.5
7	mlp_t	92.1	17	Bagging_REPTree_w	90.4
8	LibSVM_w	91.7	18	MAB_MLP_w	90.4
9	RRF_t	91.4	19	elm_m	90.3
10	dkp_C	91.4	20	rda_R	90.3

In the same way, we use Bagging, Boosting, GLM to rank the classifiers according to their accuracy and other parameters.

Summary:

This is a study of the behavior of 179 classifiers when tested for 121 data sets that are from the real world problems. The best results are achieved by using ParRF_t which provides 94.1% accuracy over all datasets and overcomes 90% accuracy in 102 out of 121 datasets. Thus, ParRF_t may be used as a reference to compare the other classifiers with. Best performing classifiers are tuned in R which may be used for such problems.

Complete reference: **Customer churn prediction using improved balanced random forests**; Y Xie, X Li, EWT Ngai, W Ying; Expert systems with applications [0957-4174] Xie yr: 2009 vol: 36 iss: 3 pg: 5445 -5449

Reviewed by: Sushant Chittoor Ravinder

Objective:

In many real world applications like, for example, churn prediction, imbalance in data distribution is a very relevant and demanding problem. The key problem this paper is addressing is how to deal with imbalance in data distribution. In this paper, the authors propose a new method called Improved Balanced Random Forest (IBRF) method, obtained by combining Balanced Random Forests and Weighted Random Forests, to predict churn in the banking industry. According to the authors, this method helps address the problem of imbalance in data by altering the class distribution and by putting higher penalties on misclassification of the minority class.

Approach:

The authors combine the sampling techniques of balanced random forest and the cost-sensitive learning of weighted random forest to change the class distribution and penalize more heavily the misclassification of the minority class. To combine these two methods, they introduce two ‘‘interval variables” m and d, where m is the middle point of an interval and d is the length of the interval.

The algorithm used by the authors takes a training set as input and splits it into two subsets with one set containing all the positive samples while the other set having all the negative samples. For t=1,2...n_{tree}, a variable α is randomly generated within the interval m - d/2 and m + d/2. nα sample is drawn with replacement from the negative training dataset and n(1-nα) sample is drawn with replacement from the positive training dataset. An unpruned classification tree is grown.

All test samples are ordered by the negative score of each sample. The negative score of each sample is considered to be the total number of trees which predict the sample to be negative. The more trees predict the sample to be negative, the higher negative score the sample gets.

Results:

IBRF is better classifier as its lift curve is higher than the other algorithms.

Table 2
Experimental results of different algorithms

Algorithm	ANN	DT	CWC-SVM	IBRF
Accuracy rate	78.1%	62.0%	87.2%	93.2%
Top-decile lift	2.6	3.2	3.5	7.1

ANN, artificial neural network; DT, decision tree; IBRF, improved balanced random forests.

Summary:

The major takeaways from this work are as follows:

- The authors combine the sampling techniques of balanced random forest and the cost-sensitive learning of weighted random forest to change the class distribution and penalize more heavily the misclassification of the minority class.
- According to the study, the best features are iteratively learned by artificially making class priors equal, based on which best weak classifiers are derived.
- According to the study, IBRF produces higher accuracy than other random forests algorithms such as balanced random forests and weighted random forests

Also, IBRF employs internal variables to determine the distribution of samples. Although the results are found to be insensitive to the values of these variables, imposing some limitations on them in future experiments may enhance the predictive effectiveness of the method. In addition, there is further research potential in the inquiry into the cost-effectiveness of this method. **This is another paper focused on imbalanced data set specially for customer churn prediction, this paper provide details on how to apply cost sensitive weightage to balance the results, which could be helpful in our project.**

Reference: **Global Optimization Ensemble Model for Classification Methods**; Hina Anwar, Usman Qamar, Abdul Wahab Muzaffar Qureshi; Vol. 2014, pp. 313164. Date of Electronic Publication: 2014 Apr 27.

Reviewed by: Sushant Chittoor Ravinder

Objective:

This paper proposes a global optimization ensemble model for classification methods (GMC) that can improve the overall accuracy for supervised learning problems. Some basic issues like bias-variance tradeoff, dimensionality of input space, and noise in the input data space, that affect the accuracy of a classifier, are dealt with in this paper. The authors designed the GMC in layers with each layer solving one of the basic issues mentioned.

Approach:

The authors designed the global ensemble model by carrying out operations in different layers as shown in the figure below. In each of the layers, the following actions are executed.

- **Layer 1:** Genetic Algorithm is used for feature selection
- **Layer 2:** Partition of data is done using X- Fold Cross Validation
- **Layer 3:** Bagging is done to obtain the optimal bias-variance trade-off.
- **Layer 4:** Classifiers are placed in this layer.

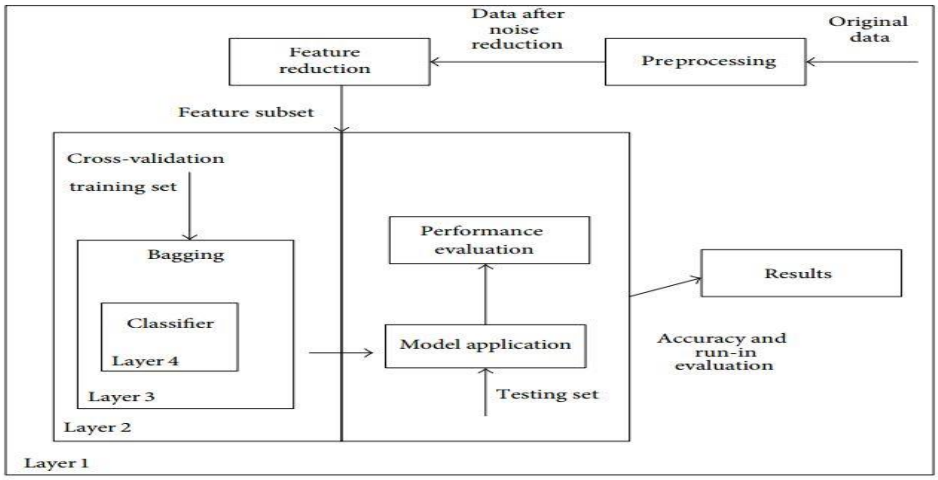


FIGURE 1: Design of global optimization ensemble model for classification methods (GMC).

This approach helped the authors achieve greater accuracies when used on various public datasets.

Results:

The authors used their model on 7 different datasets, details and results of which are given below.

TABLE 6: Results for cancer dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	66.81%	96.57%	29.76%
Decision tree	94.42%	96.71%	2.29%
ID3	66.52%	85.27%	18.52%
W-PART	94.71%	97.28%	2.57%
W-Prism	90.13%	96.28%	6.15%
W-J48	94.71%	96.71%	2%
W-AODE	97.00%	100%	3%
Logistic regression	95.01%	96.14%	1.13%

TABLE 7: Results for heart disease dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	50.82%	59.75%	8.93%
Decision tree	44.89%	59.43%	14.54%
ID3	47.52%	55.48%	8.24%
W-PART	50.52%	60.08%	9.56%
W-Prism	47.51%	56.09%	8.58%
W-AODE	55.47%	61.13%	5.66%
W-J48	49.87%	61.05%	11.18%
Rule induction	57.72%	59.76%	2.4%

TABLE 8: Results of wine dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	70.75%	90.42%	19.67%
Decision tree	91.57%	95.49%	3.92%
W-PART	90.42%	96.67%	6.25%
W-Prism	52.32%	61.27%	8.95%
W-AODE	71.34%	75.26%	3.92%
W-J48	90.46%	96.63%	6.17%
Rule induction	86.37%	93.27%	6.9%

TABLE 9: Results of adult income dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	76.70%	83.20%	6.5%
Decision tree	80.00%	82.20%	2.20%
ID3	75.60%	78.60%	3%
W-PART	81.00%	83.50%	2.4%
W-Prism	81.10%	82.20%	1.1%
W-AODE	80.80%	82.60%	1.8%
W-J48	81.50%	83.00%	1.5%
Random forest	76.10%	77.30%	1.2%
Logistic regression	79.00%	80.00%	1%

In this study, the experimental results on various public datasets showed that the proposed model improved the accuracy of the classification models from 1% to 30% depending upon the algorithm complexity. This implies that the GMC used by the authors yielded better results than when the classifiers were used alone or in combination. Some basic issues of supervised learning problems like dimensionality reduction, bias-variance tradeoff and noise can be countered by using the concept of ensemble models to design an optimized global ensemble model (GMC). **The layer 2 discussion on x-fold cross validation will be applied to our model.**

Project approach

As briefly discusses in the introduction, the acquired data is assessed for various data discrepancies. These discrepancies are addressed through application of various data cleaning and filtering techniques in the pre-processing stage. This includes removal of data points with erratic readings, removal of highly correlated

predictors, removal of variables that are linear combination of other variables, duplicated columns and rows. This pre-processing step is followed by popular subset selection based feature selection methods. The best performing feature selection method obtained by calculating AUC on a logistic regression model is selected for rest of the analysis. In order to eliminate majority class bias in our modelling step, oversampled data is used to train our model. This trained model is tested on imbalanced original data separated before oversampling. A flow chart of the process is given in Figure 1.

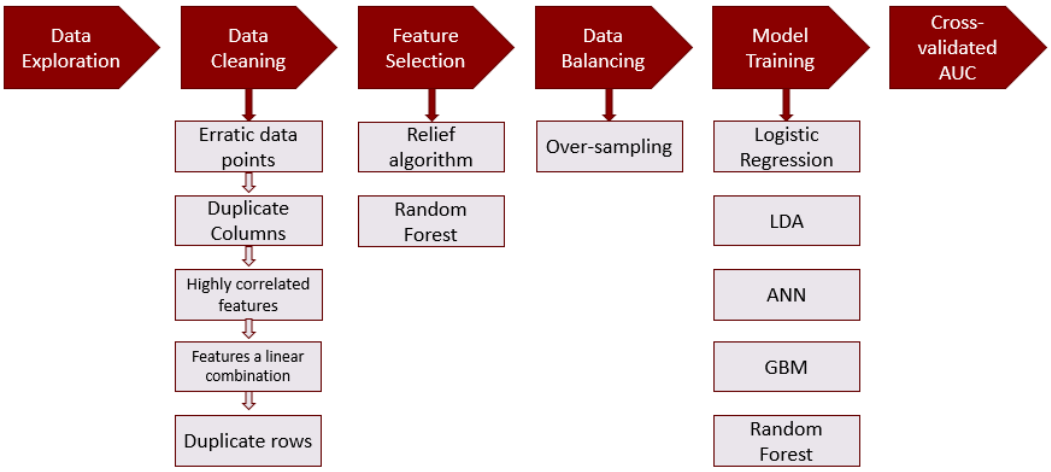


Figure 1 Overall flow chart of the approach taken

As learnt in class, for any model to be reported cross validation needs to be performed. A 10 fold cross validation loop is implemented in this case. This loop divides the data into 10 folds, care is taken that the representation of the minor class in all of the folds are representative of the original dataset. In the first loop, one of the folds is taken as test set and model is trained on the remaining 9 folds. The data of the training 9 folds are balanced using oversampling before training the model. This process is then repeated for all the ten folds one by one and tested on corresponding nine fold trained model. This process is described in the Figure 2.

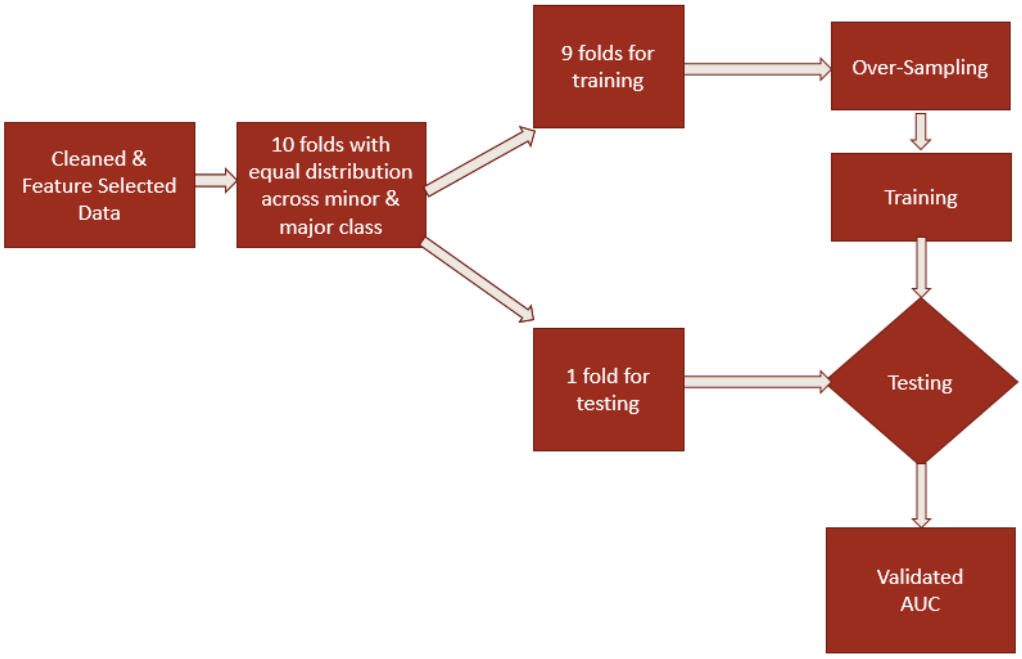


Figure 2. Ten Fold cross validation for the dataset

New Techniques

Since the original data was of relatively higher dimension than what we studied in class, it presented with major challenges in implementation. The original data had noise and repetitive variables that resulted in frequent errors. The sheer dimension of the data resulted in hours of single code running (6.6 hours for relief feature selector). Given the constraint of time, following new techniques were applied to overcome these challenges.

Pre-processing of data: Various filter functions were employed to clean the dataset.

Relief function: Relief function in FSelector library, is a popular weight assigning subset selection method for binary classification proposed by Kira and Rendell. It is relatively suitable for noisy data and robust in feature interactions, however may not be good for correlated features. Therefore, a higher correlation cut percentage (80%) was applied as a pre-step to implementation of relief function. This lead to a total of 32 relevant predictors with attribute importance greater than zero.

Oversampling: In order to balance the dataset, oversampling technique was employed. It duplicates the minor class to the level, where an approximate number of minor and major class data points exist in the data. This leads to an unbiased modelling which is critical to our current problem. Oversampling was preferred over under-sampling as it leads to no loss in information.

Artificial neural network: nnet library in R was used to apply feed-forward neural network with a single hidden layer. Tuning of the parameters was carried out to improve the results.

Gradient Boosting Methods: In our search for faster and powerful classification algorithms, we came across Gradient boosting techniques. It's an ensemble method which adds new weak base learner model

sequentially in a stage wise fashion. They are fast and ideal for handling bigger data. Two of the packages used for our modelling purposes

Xgboost library includes tree learning algorithm which are almost 10 times faster than other gb packages.

H2O library which has parallel computation capabilities for big data.

Implementation details

Cleaning of data

1. Data exploration was carried out and the data was checked for missing values. No missing values were found.
2. 1310 erratic values (-999999 & 9999999999) were found in several data points. These values are suspected to be coded versions of missing values. These data points were removed from further analysis.
3. 70 predictors with constant 0 value across all observations were removed.
4. 26 duplicate predictors were removed.
5. A filtering technique was applied to remove 57 variables which are linear functions of other variables.
6. 48 correlated variables were removed after applying a cut-off factor of 0.95.
7. 4931 duplicate rows were removed.

Feature selection

1. Random Forest and Relief function (FSelector library) were applied to identify most significant variables in the cleaned data set.
2. Relief function resulted in a total of 32 predictors with attribute importance value greater than zero. A cutoff point was defined for random forest to select the top 70 most important predictors.
3. Both sets of variables obtained were tested on a logistic regression model, and the method with the best AUC was then used for further analysis. In our analysis, random forest with top 70 predictors outperformed relief by a narrow margin.

10-Fold Division of dataset & its testing on models

1. The cleaned data was divided into 10 folds, class distribution of each fold was kept constant.
2. Of the 10 folds 9 were used for training data and remaining 1 for testing the built algorithm. This was performed in 10 different combinations.
3. The training data was over-sampled to remove major class bias in our modeling.
4. Training of the model was carried out on the over-sampled data.
5. The model was then tested on the imbalanced test set. ROC analysis was performed to report AUC for each of the combinations.
6. Different modeling techniques were applied and the best performing prediction algorithm was chosen as the final model.

The complete R-code for the above mentioned implementation has been attached in APPENDIX-1.

Results

A summary of the final results is presented in the Table 1. Area under curve of the ROC is used as a measure of performance because classification rate is misleading in imbalanced data.

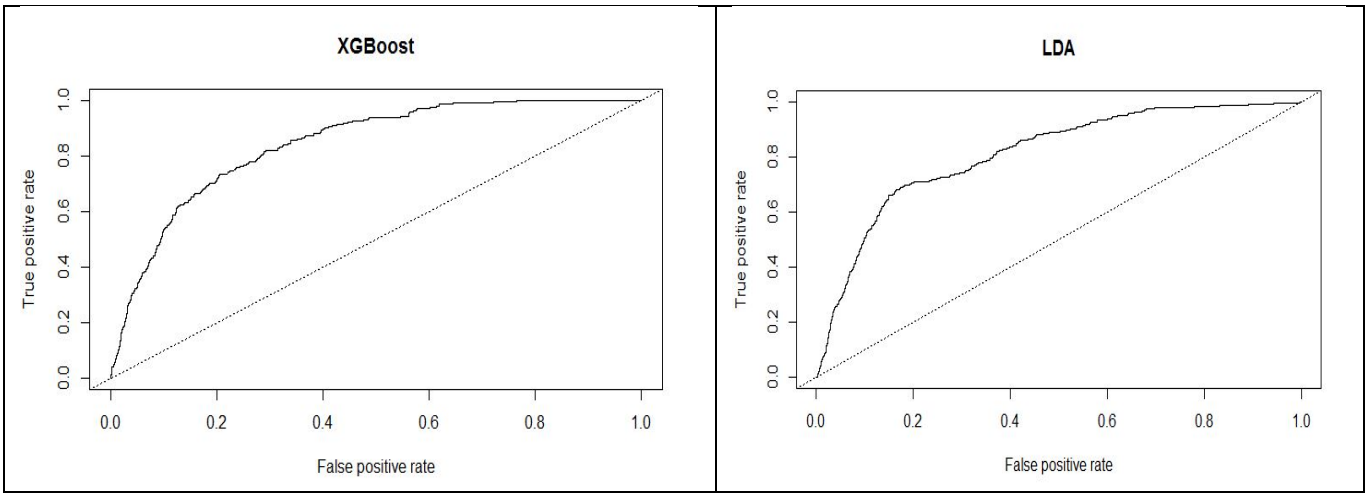
Table 1 Summary of results AUC for all applied models.

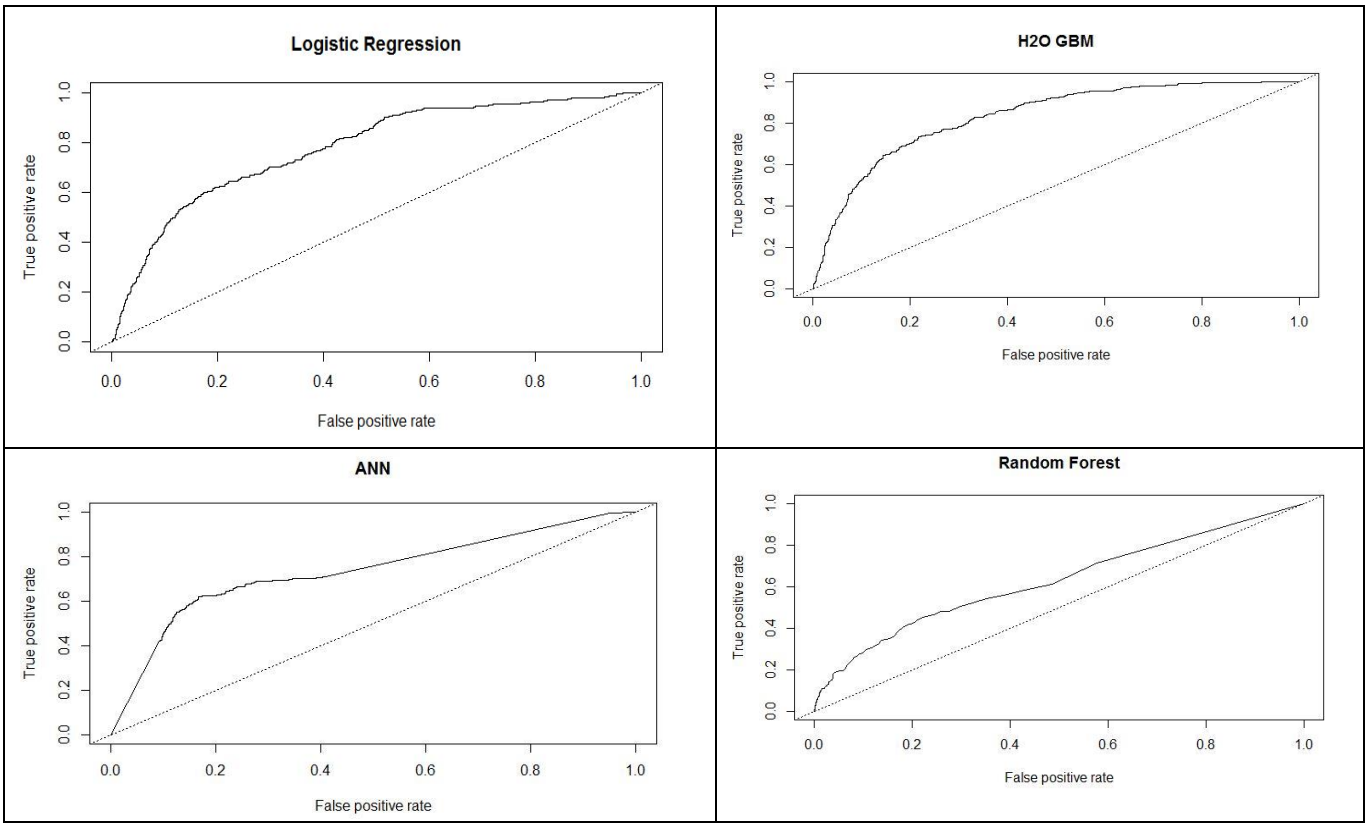
Model	10-fold Cross Validated AUC	
	Original Data	Over Sampled Training Data
Logistics Regression	0.7921	0.7958
LDA	0.788	0.7902
XG Boost	0.8416	0.8344
ANN	0.7307	0.6887
Random Forest	0.6283	0.6206
H20 GMB	0.8328	0.8264

The XGBoost gave the best AUC, with the original imbalanced dataset. The H20 GBM model also gave a very good comparable AUC. The random forest reported the lowest performance, which can be improved by tuning the model more rigorously. With LDA and Logistics regression, it can be noted that there is a slight improvement in AUC when the model is trained using oversampled data. With the ensemble models like random forest, XGBoost and ANN, a drop in performance with oversampled data can be observed.

It was further noticed that the methods like GBM, logistic regression and LDA provided a superior computational speeds. ANN and random forest method were computationally intense and took a longer period to run, hence couldn't be fine-tuned easily.

ROC plot of the individual methods are shown in the following diagram.





Conclusions

- The modern advanced techniques like gradient boosting ensemble methods XGBoost and H20 GBM are out performing the classical methods, both in terms of prediction performance and computational time.
- Oversampling was found useful to improve the classical methods like Logistics regression and LDA. But the modern sophisticated models performed better on the original imbalanced data.
- The quality of and the size of the dataset is important for the models like logistics Regression, LDA and random forest. The XGBoost ,h2o.gbm and ANN was performing similarly with feature selected and all feature datasets.

Individual Roles & Responsibilities

It's a collaborative project and required review and participation of all members in each of the steps involved. However, individual focus areas are divided as follows:

Adithya Ajith: Cross validating each models, modeling new techniques like ANN and ensemble models.

Muzzamil Bashir: Data cleaning, variable selection by using Fselector and random forest.

Nirav Thaker: Generating ROC and finding AUCs. Analysis and report generation.

Sushant Chittoor Ravinder: Logistic regression, LDA and random forest modelling.

References

https://h2o-release.s3.amazonaws.com/h2o/rel-slater/9/docs-website/h2o-docs/booklets/GBM_Vignette.pdf
https://cran.r-project.org/web/packages/nnet/nnet.pdf
https://cran.r-project.org/web/packages/h2o/h2o.pdf
http://xgboost.readthedocs.io/en/latest/model.html
https://cran.r-project.org/web/packages/xgboost/xgboost.pdf

Appendix-I

R-CODE

```
train = read.csv("train.csv")      # load data
train$ID = NULL
#Removing ID as it plays no role in the machine learning process
dim(train)

#Removing erratic values
train[train==-999999] <- train[train==9999999999] <- NA
sum(is.na(train))
train <- na.omit(train)
sum(is.na(train))

train.y = as.factor(train$TARGET); train$TARGET = NULL  # Extracting the
response - TARGET#

###Cleaning the dataset - Part_1
# Removing 70 predictors having constant 0 value across all observations
for (f in setdiff(names(train),c('TARGET'))) {
  if (mean(train[[f]])== sum(train[[f]])) {
     #cat(f, "is constant in train.\n")
    train[[f]] = NULL
      }
}
rm(f)

library(digest)
# Removing 26 duplicated predictors.
dup = duplicated(lapply(train, digest))
train = train[!dup]
rm(dup)

# 0 count per row
count0 <- function(x) return( sum(x == 0) )
train$n0 = apply(train, 1, FUN=count0)
#making a new feature, thats the count of the number of zeros in the row
#boxplot(train$n0~train.y ,main= "Count of Zeros vs Customer Satisfaction")

library(caret)
# Removing features being a specific linear function of other features
lin.comb = findLinearCombos(train)
train = train[, -lin.comb$remove];  rm(lin.comb)

# Removing the correlated features
cor.features = findCorrelation(cor(train), cutoff = .95, verbose = FALSE)
train = train[, -cor.features];        rm(cor.features)

#Removing duplicate rows
dup=!duplicated(train);
train.y = train.y[dup]
```

```
train= train[dup,] ; rm(dup)# removes duplicated rows
dim(train)
table(train.y)

# Ranking features by importance
if(F)  #put T , to run the following chuck
{
#For each feature, the % of rows that have the most frequent value.
feature <- most_freq_percent <- numeric(length(names(train)))
for (i in 1:length(names(train))) {
  tabl = as.data.frame(table(train[[names(train)[i]]]))
  tabl = tabl[order(tabl$Freq,decreasing=TRUE),]
  #cat(names(train)[i] ,"\n")
  feature[i] = names(train)[i]
  most_freq_percent[i] = round(100.0 * tabl[1,2] / nrow(train),5)
}
percent= data.frame(feature , most_freq_percent ); rm(feature ,
most_freq_percent)
write.csv(file="unique_importance.csv", x=percent , row.names = T)

train$TARGET = train.y; rm(train.y)

# Chi Test to find the ranking of the variables.
chi_weights <- chi.squared(TARGET~., train)
write.csv(file="Chi_squared_importance.csv", x=importance , row.names = T)

# Relief algorithim to find the ranking of the variables.
system.time(relief_weights <- relief(TARGET~., data=train, neighbours.count =
5, sample.size = 5))
write.csv(file="Relief_weights_importance.csv", x=importance , row.names = T)

# Random Forest to find the ranking of the variables.
system.time(forest <- randomForest(TARGET ~ .,data=train , importance =TRUE))
importance <- forest$importance
write.csv(file="Random_forest_importance.csv", x=importance , row.names = T)
varImpPlot(forest)}

#importing feature names with importance
fsel=read.csv("Name_importance.csv" , row.names = 1)
name=row.names(fsel[fsel$importance>1,]);  rm(fsel)

train = train[,c(name)] ;rm(name)  # feature selection based on the ranking

# removes duplicated rows
dup=!duplicated(train)
train.y = train.y[dup]
train= train[dup,] ; rm(dup)
dim(train)
table(train.y)

train$TARGET =train.y; rm(train.y)
#Jumbling the order of the rows in the Dataset and this data set will be used
for CV purposes
set.seed(1);
```

```
train = train[sample(nrow(train)),]
train_0 = train[train$TARGET== 0,]
train_0$folds <- cut(1:nrow(train_0),breaks=10,labels=FALSE)  # the 10 folds to
be used for the 10-fold cross validation throughout this report.
train_1 = train[train$TARGET==1,]
train_1$folds <- cut(1:nrow(train_1),breaks=10,labels=FALSE)

train = rbind(train_0,train_1); rm(train_0 , train_1)
folds = train$folds; train$folds =NULL
table(folds)

library(ROCR)
#Function that returns ROC plot and the AUC
ROCplot =function(probabilty ,test_y , ...){
predob = prediction(probabilty , test_y)
perf = performance (predob , measure = "tpr", x.measure = "fpr")
auc= performance(predob, measure = "auc")
plot(perf,...)
abline(a=0, b= 1, lty=3)
return(round(auc@y.values[[1]],3))
}
####################
#### MODELING ######
####################

#Logistic Regression
#without sampling
CV_AUC_LG = numeric(10)
for(i in 1:10){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    pred.model = glm( TARGET ~  . , data = train[-testIndexes,], family =
"binomial")
    buffer = predict(pred.model , train[testIndexes,] , type = "response")
    CV_AUC_LG[i] = ROCplot(buffer , train$TARGET[testIndexes],main ="Logistic
Regression")
}
rm(buffer,pred.model,testIndexes,i)
mean(CV_AUC_LG)

# with oversampling
CV_AUC_LG_oversampling = numeric(10)
for(i in 1:10){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    pred.model = glm( Class ~  . , data = up_train, family = "binomial")
    buffer = predict(pred.model , train[testIndexes,] , type = "response")
    CV_AUC_LG_oversampling[i] = ROCplot(buffer , train$TARGET[testIndexes],main
="Logistic Regression")
}
```

```
rm(buffer,up_train,pred.model,testIndexes,i)
mean( CV_AUC_LG_oversampling )

#LDA
library(MASS)

# without sampling
CV_AUC_LDA = numeric(10)
for(i in 1:10){
  cat(i)
    #Segment your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    pred.model = lda( TARGET ~  . , data = train[-testIndexes,])
    buffer = predict(pred.model , train[testIndexes,])
    CV_AUC_LDA[i] = ROCplot(buffer$posterior[,2] ,
train$TARGET[testIndexes],main ="LDA")
}
rm(buffer,testIndexes,pred.model,i)
mean( CV_AUC_LDA)

# with over-sampling
CV_AUC_LDA = numeric(10)
for(i in 1:10){
  cat(i)
    #Segment your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    pred.model = lda( Class ~  . , data = up_train[-testIndexes, ])
    buffer = predict(pred.model , train[testIndexes,] )
    CV_AUC_LDA_oversampling[i] = ROCplot(buffer$posterior[,2] ,
train$TARGET[testIndexes],main ="LDA")
}
rm(buffer,up_train,pred.model,i)
mean( CV_AUC_LDA_oversampling)

#Random Forest
library(randomForest)
# train
CV_AUC_rf = NULL
for(i in 1:10){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    set.seed(1)
    system.time(pred.model <- randomForest(train[-testIndexes,-ncol(train)],
train$TARGET[-testIndexes] ))
    buffer = predict( pred.model , newdata = train[ testIndexes,] , type=
"prob")
    CV_AUC_rf[i] = ROCplot(buffer[,2] , train$TARGET[testIndexes],main ="Random
Forest")
}

mean(CV_AUC_rf)
```

```
# with over sampling
CV_AUC_rf_oversampling = NULL
for(i in 1:3){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    set.seed(1)
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    pred.model = randomForest(Class ~ ., data= up_train )
    buffer = predict( pred.model , newdata = train[ testIndexes,] , type=
"prob")
    CV_AUC_rf_oversampling[i] = ROCplot(buffer[,2] ,
train$TARGET[testIndexes],main ="Random Forest")
}

mean(CV_AUC_rf_oversampling)

##XGBoost

library(xgboost)
#without sampling
CV_AUC_XG = NULL
xg_train = train
xg_train$TARGET <- as.numeric(levels(xg_train$TARGET))[xg_train$TARGET]
for(i in 1:10){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    train_xg <- sparse.model.matrix(TARGET ~ ., data = xg_train[-testIndexes,])
    dtrain <- xgb.DMatrix(data=train_xg, label=xg_train$TARGET[-testIndexes])
    watchlist <- list(train_xg=dtrain)
    param <- list( objective = "binary:logistic",
               booster = "gbtree",
               eval_metric = "auc",
               eta = 0.0202048,
               max_depth = 5,
               subsample = 0.6815,
               colsample_bytree = 0.701)
    clf <- xgb.train( params = param,
                  data = dtrain,
                  nrounds = 560,
                  verbose = 1,
                  watchlist = watchlist,
                  maximize = FALSE)
    test_xg <- sparse.model.matrix(TARGET ~ ., data = xg_train[testIndexes,])
    preds <- predict(clf, test_xg)
    CV_AUC_XG[i] = ROCplot(preds , xg_train$TARGET[testIndexes],main
="XGBoost")

}
rm(test_xg,param,i,preds,watchlist,dtrain,train_xg,testIndexes,xg_train)
mean(CV_AUC_XG)
```

```
#with over sampling
CV_AUC_XG_oversampling = NULL
for(i in 1:10){
  cat(i)
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    up_train$Class <- as.numeric(levels(up_train$Class))[up_train$Class]
    train_xg <- sparse.model.matrix(Class ~ ., data = up_train)
    dtrain <- xgb.DMatrix(data=train_xg, label=up_train$Class)
    watchlist <- list(train_xg=dtrain)
    param <- list( objective = "binary:logistic",
               booster = "gbtree",
               eval_metric = "auc",
               eta = 0.0202048,
               max_depth = 5,
               subsample = 0.6815,
               colsample_bytree = 0.701)
    clf <- xgb.train( params = param,
                  data = dtrain,
                  nrounds = 560,
                  verbose = 1,
                  watchlist = watchlist,
                  maximize = FALSE)
    test_xg <- sparse.model.matrix(TARGET ~ ., data = train[testIndexes,])
    preds <- predict(clf, test_xg)
    CV_AUC_XG_oversampling[i] = ROCplot(preds , train$TARGET[testIndexes],main
="XGBoost")

}
rm(test_xg,param,i,preds,watchlist,dtrain,train_xg,testIndexes,up_train)

mean(CV_AUC)
CV_AUC_XG_oversampling


## ANN
library(nnet)
#without sampling
CV_AUC_ANN = NULL
for(i in 1:10){
    cat(i)
    testIndexes <- which(folds==i,arr.ind=TRUE)
    set.seed(1)
    ideal <- class.ind(as.factor(train$TARGET))
    ANN = nnet( x= train[-testIndexes,-ncol(train)], y= ideal[-testIndexes,],
size=5,maxit=200, decay=5e-2)
    pred = predict(ANN, train[testIndexes,-ncol(train)], type="raw")

    CV_AUC_ANN[i] = ROCplot(pred[,2] , train$TARGET[testIndexes],main ="ANN")
}
```

```
rm(pred,ideal,i,testIndexes,ANN)
mean(CV_AUC_ANN)

#with oversampling
CV_AUC_ANN_oversampling = NULL
for(i in 1:10){
    cat(i)
    testIndexes <- which(folds==i,arr.ind=TRUE)
    set.seed(1)
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    ideal <- class.ind(as.factor(up_train$Class))
    ANN = nnet( x= up_train[,-ncol(up_train)], y= ideal, size=5,maxit=200,
decay=5e-2)
    pred = predict(ANN, train[testIndexes,-ncol(train)], type="raw")

    CV_AUC_ANN_oversampling[i] = ROCplot(pred[,2] ,
train$TARGET[testIndexes],main ="ANN")
}
rm(pred,ideal,up_train,i,testIndexes,ANN)
mean(CV_AUC_ANN_oversampling)

##H2O GBM

library(h2o)
h2o.init(nthreads=-1)
col <- colnames(train)[-ncol(train)]

#without sampling
train.hex <- as.h2o(train, destination_frame = "train.hex")
CV_AUC_h2Ogmb = NULL
for(i in 1:10){
    cat(i)
    testIndexes <- which(folds==i,arr.ind=TRUE)
   h2o.gbm <- h2o.gbm(y = "TARGET", x = col, training_frame = train.hex[-
testIndexes,],
                           ntrees = 500, max_depth = 3, min_rows = 2)
    h2o.gbm.pred = h2o.predict(object = h2o.gbm, newdata =
train.hex[testIndexes,])

    h2o.gbm.pred.df <-as.data.frame(h2o.gbm.pred)
    CV_AUC_h2Ogmb[i] = ROCplot(h2o.gbm.pred.df [,3] ,
train$TARGET[testIndexes],main ="H2O GBM")
}
rm(h2o.gbm,h2o.gbm.pred,h2o.gbm.pred.df,i,testIndexes,train.hex)
mean(CV_AUC_h2Ogmb)

#with oversampling
CV_AUC_h2Ogmb_oversampling = NULL
test.hex <- as.h2o(train, destination_frame = "test.hex")
for(i in 1:10){
    cat(i)
    testIndexes <- which(folds==i,arr.ind=TRUE)
```

```
    up_train = upSample(x = train[-testIndexes,-ncol(train)],y = train$TARGET[-
testIndexes])
    train.hex <- as.h2o(up_train, destination_frame = "train.hex")
    h2o.gbm <- h2o.gbm(y = "Class", x = col, training_frame = train.hex,ntrees
= 500, max_depth = 3, min_rows = 2)
    h2o.gbm.pred = h2o.predict(object = h2o.gbm, newdata =
test.hex[testIndexes,])

    h2o.gbm.pred.df <-as.data.frame(h2o.gbm.pred)
    CV_AUC_h2Ogmb_oversampling[i] = ROCplot(h2o.gbm.pred.df [,3] ,
train$TARGET[testIndexes],main ="H2O GBM")
}

rm(h2o.gbm,h2o.gbm.pred,h2o.gbm.pred.df,i,testIndexes,train.hex,test.hex,up_tra
in)
mean(CV_AUC_h2Ogmb_oversampling)
```

APPENDIX-II

FALL 2016 ISEN 613 Project's

Literature Review

By Team 13

Reviewed by: Adithya Ajith

Reference: Haibo He and Edwardo A. Garcia, "**Learning from Imbalanced Data**" IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 9, September 2009

Objective:

a) The key problem addressed in the paper.

The challenges for learning from imbalanced data is reviewed in this paper. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms, which assume or expect balanced class distributions or equal misclassification costs.

b) Importance of addressing this problem.

As most of the real world data being generated are imbalanced, the problem of learning from such data has attracted attention from both academia and industry. Due to the unbalanced class distributions, the imbalanced learning problem is concerned with the performance of most standard learning algorithms.

c) The key objective of this paper addresses.

To provide a broad survey and review of the development of research in learning from imbalanced data. Also, to develop a comprehensive resource that can be used by data engineering researchers and practitioners.

d) Methodology of addressing this problem.

A critical review of the nature of the problem, the solutions or methods, and the current assessment metrics used to evaluate learning performance under the imbalanced learning scenario has been made.

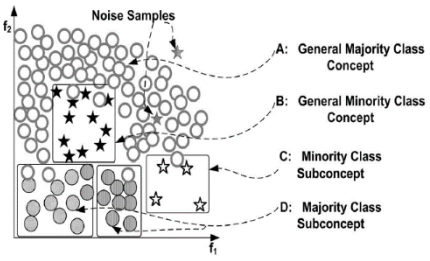
e) Findings and its implication from the paper

Even though unbalanced dataset have inherent issues with the popular learning methods, the methods discussed in this paper will provide a researcher or a practitioner with resources to tackle the problem.

Approach:

By doing an extensive review of the literature available on the topic, the authors have made an attempt to build a toolbox that can be used in machine learning of imbalanced data.

The authors first elucidates the nature of the imbalanced data. It is shown how conventional evaluation practice of using assessment criteria, such as the overall accuracy or error rate, does not provide adequate information in the case of imbalanced learning. Different types of imbalances - intrinsic and extrinsic, then relative imbalance and imbalance due to rare instances are explained. Various research suggests that apart from the between class imbalance, data complexity (issues such as overlapping, lack of representative data, small disjuncts, etc) is the primary determining factor of classification deterioration. This data complexity leads to within-class imbalance, as shown in the figure.



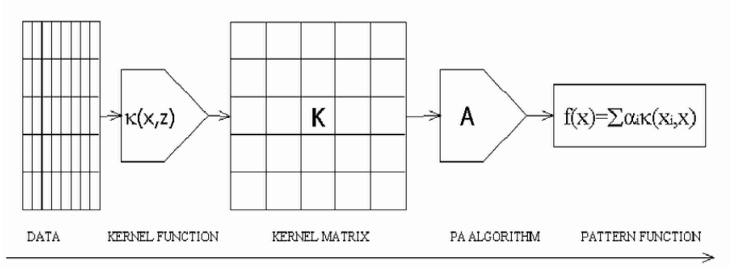
A high-complexity data set with both between-class and within-class imbalances, multiple concepts, overlapping, noise, and lack of representative data.

The various available solutions for imbalanced learning are cited and explained in the paper as listed below:

Sampling Methods: Sampling technique refers to the different mechanisms that transforms an imbalanced data to have a balanced class distribution. The different methods popular in the research community that are discussed in the paper are : random oversampling and undersampling, informed undersampling, synthetic sampling with data generation, adaptive synthetic sampling, sampling with data cleaning techniques, cluster-based sampling method and integration of sampling and boosting.

Cost-Sensitive Methods: As an alternative to sampling methods, cost-sensitive learning targets the imbalanced learning problem by using different penalties that describe the costs for misclassifying any particular data example. The different methods discussed in the paper are: cost-sensitive learning framework, cost-sensitive dataspace weighting with adaptive boosting, cost-sensitive decision trees and cost-sensitive neural networks

Kernel-Based Methods: Kernel methods offer a modular framework for machine learning. First, a dataset is processed into a kernel matrix. Then, a variety of kernel algorithms can be used to analyze the data, using only the information contained in the kernel matrix. The methods discussed includes integration of kernel methods with sampling methods – using SVM and Granular Support Vector Machines, kernel modification methods and active Learning Methods with kernel-based learning methods.



Source: J. Shawe-Taylor and N. Cristianini, Kernel methods for pattern analysis, 2004

The authors briefly describes One-class learning and Mahalanobis-Taguchi System that can be used to classify unbalanced dataset.

After discussing all the methods used to train a learning model from the imbalanced data, the authors stresses the importance of standardized evaluation metrics to properly assess the effectiveness of each model. The different metrics used across the research community are: singular assessment metrics (precision, recall, F-measure, and G-mean), receiver operating characteristics (ROC) curves, precision-recall (PR) curves and cost curves. Authors ends this discussion by reviewing the assessment metrics for multiclass imbalanced learning

Summary:

Major takeaway of this paper is that the discussions about the nature of the imbalanced learning problem, the methodologies, and the several major assessment techniques that can be used to evaluate the problem of learning from unbalanced data, will aid as a comprehensive resource for learning from unbalanced dataset.

Reviewed by: Adithya Ajith

Reference: Stephan Dreiseitl and Lucila Ohno-Machado , "**Logistic regression and artificial neural network classification models: a methodology review** " Journal of Biomedical Informatics,Volume 35, Issues 5–6, October 2002, Pages 352–359, Science Direct

Objective:

a) The key problem addressed in the paper.

Determining the factors to be considered when judging research results using the logistic regression and neural network models

b) Importance of addressing this problem.

Since mistakes in modeling and evaluation can have disastrous consequences, care must be taken to ensure that the models are validated and evaluated correctly.

c) The key objective of this paper addresses.

The paper is focused on understanding the differences and similarities of Logistic Regression and Artificial Neural Network Classification models, in comparison to other machine learning models. Also to determine factors need to be considered when judging research results using predictive models.

d) Methodology of addressing this problem

The authors make use of available research journals on predictive modelling in the field of medical domains to compare logistic regression and artificial neural network models. Analyze its performance and popularity as compared to the other modelling techniques such as Decision trees, KNN and Support Vector Machines.

e) Findings and its implication from the paper

By analyzing the model with respect to criteria such as data set size and performance measure, the authors point out which factors need to be considered when judging research results using predictive models. These findings can be deployed in the future predictive modeling research.

Approach:

Comparison of features of SVM, Logistic regression, ANN, KNN and Decision trees is done and its respective advantages and disadvantages listed. The authors point out that SVM is build separating boundaries by solving quadratic optimization problem and since varying degree of nonlinearity and flexibility can be included in the model. The disadvantage of SVM being the classification result is dichotomous, and no probability of class membership is found out. In KNN, the only adjustable parameter being the number of closest neighbors to estimate the class membership, there is not much information on model construction, but they can provide good explanation for the classification result. Decision trees utilize the criterion of information gain while modeling and that they are easily interpretable although this modelling deploys a greedy approach.

Logistic regression and ANN differ from KNN,SVM and Decision trees in the sense that they provide functional forms and parameter details by maximum likelihood estimation, the difference being the first is parametric and the latter semi parametric /non parametric. In Logistic regression, there is scope for including interaction and non-linear terms making the model flexible at the expense of overfitting. The functional form of ANN is significantly different from Logistic regression, and due to the

non-linearity of hidden neurons the model is more flexible than logistic regression. In the Parameter estimation technique, logistic regression being less complex when no non-linear or interaction terms are used reduces the risk of overfitting. To restrict the complexity of neural networks, the author suggests using regularization called weight decay early stopping. The Bayesian framework is believed to be an alternative to solve problem of overfitting.

With respect to variable selection, logistic model is more popular due to the simplicity of modelling with forward, backward and step selection. For neural networks, automatic relevance discrimination or sensitivity analysis is used to assess the importance of predictors.

The trained model can be evaluated by using the criteria of discrimination and calibration. Discrimination is a measure of how well the two classes in the data set are separated and calibration determines how accurate the model class probability estimate compared to the true probability. Validation, cross validation or bootstrap is used in evaluating these criteria. Common metrics of discrimination are sensitivity, specificity, accuracy and the area under the ROC curve. In order to calibrate, difference between the average observation and the average outcome of a given group is measured.

The authors further analyzed the 72 papers with respect to the following criteria: whether details of the model building process are given (variable selection scheme for logistic regression, parameter selection and overfitting avoidance for artificial neural networks), whether unbiased estimates of the generalization error are reported (by using test sets, cross-validation, or bootstrapping), whether measures of discriminatory power were given (and statistical testing using these measures), and whether calibration information is included. Also the performance comparison of both the models were made.

Results:

The authors on reviewing available literature and papers using both logistic regression and neural network for predictive modelling come to a conclusion that both models perform at about the same level with the more flexible neural networks generally outperforming logistic regression in the remaining cases.

ANN better	LR better	No difference
51 %	7 %	42 %

Table 1: Summary of comparing the discriminatory power of artificial neural networks with logistic regression models, as percentage of 72 papers

The authors arrive at the conclusion that white box models such as Decision trees, Logistic regression offer better interpretability whereas black box models such as SVM and Neural networks have better predictability. As per the findings, the authors conclude that the performance of logistic regression and ANN is superior to KNN and decision trees in most of the experimental cases whereas SVM shows comparable results. The popularity of logistic regression is related to its low generalization error.

Summary:

With the constant increase of importance of data analytics, several predictive algorithms are available. This paper brings out the subtle differences between the various classification models and points out their advantages and disadvantage.

Reviewed by: Nirav Thaker

Reference: Improved decision making in customer churn prediction context using generalized additive models by Kristof Coussement, Dries Benoit and Dirk Van Den Poel. http://www.sciencedirect.com/science/article/pii/S0957417409007325

Objective:

a) Key problem: Identifying risky customers in a service industry. In this case, the industry is a newspaper publishing company.
b) Key reason for the paper: To improve upon the accuracy of prediction models in churn prediction context.
c) Key objective: This paper shows that GAM is better over logistic models in these three aspects, 1) They help identify the risky customers better. 2) GAM models have better interpretability. 3) GAM models can be applied with good effect for the marketing managers to make their strategy on how to hold the risky/dissatisfied customers.
d) How did the authors address the problem: By experimenting on a dataset provided by Belgian newspaper publishing company and comparing logistic and GAM models
e) What did they find: That GAM perform better than the logistic models.
f) What does it imply: Use of GAM models is advisable and profitable in real world scenario.

Approach:

5) A methodological link between the logistic models and GAM
6) Generalized additive models are represented by replacing a term in the logistic equation with an additive term, as follows:

$$\text{logit}\{P(X)\} \equiv \log\left\{\frac{P(X)}{1-P(X)}\right\} = \alpha + \sum_{j=1}^{p} \beta_j' X_j$$

to

$$\text{logit}P(X) \equiv \log\left\{\frac{P(X)}{1-P(X)}\right\} = \alpha + \sum_{j=1}^{p} s_j(X_j)$$

7) Feature selection techniques are used: In churn modelling, a subset is selected of the original predictive variables to improve the comprehensibility.
8) The factors used to evaluate the classifiers are a) top-decile lift and b) AUC

Now, we use the above proposed methodology to real world data. The data is provided by the largest newspaper publishing company in Belgium. The response is 0 and 1 respectively, representing if they renewed the subscription within a 30 day period or not. To compare the accuracy pf logistic models and GAM, both the models are build.

Results:

To build the model, Mod_Log selects 18 predictive variables where as Mod_GAM selects 27 predictive variables. After going through the details, it is realized that the predictive terms in GAM are much more significant than they the ones in Logistic. The accuracy of GAM is better than that of Logistic models when tested.

Test set performance in terms of AUC and top-decile lift.

	AUC	Top-decile lift
MOD_LOG	0.8298[a]	4.08
MOD_GAM	0.8452[a]	4.26

[a] Significant with $\chi^2 = 187.40$, $df = 1$, $p < 0.001$.

Interpretation:

The interpretability of the GAM models is also better. The relaxation of linearity constraint allows the data scientists to have more insight into the explicative variables.

Real world applications and managerial implication:

The data shows how the company's profits are affected by using the GAM technique. The marketing analysts are interested in profits as little as 10% when they can predict the customer satisfaction rate batter. The satisfaction rate is directly resulted into company's profits and increased subscription. It is advisable to use GAM techniques to improve the power of customer churn prediction churn model.

Managerial implications.

RANDOM			MOD_LOG				MOD_GAM				
Top-decile		*Discounted profit*	*Top-decile*		*Discounted profit*	*Marginal Profit*	*Top-decile*		*Discounted profit*	*Marginal profit*	
Lift	#churners	RANDOM	Lift	#churners	MOD_LOG	MOD_LOG–RANDOM	Lift	#churners	MOD_GAM	MOD_GAM–RANDOM	MOD_GAM–MOD_LOG
1	1000	19,230.77	4.08	4080	78,461.54	59,230.77	4.26	4260	81,923.08	62692.31	3461.54
1	2000	38,461.54	4.08	8160	156,923.08	118,461.54	4.26	8520	163,846.15	125,384.60	6923.07
1	3000	57,692.31	4.08	12,240	235,384.62	177,692.31	4.26	12,780	245,769.23	188,076.90	10,384.61
1	4000	76,923.08	4.08	16,320	313,846.15	236,923.07	4.26	17,040	327,692.31	250,769.20	13,846.16
1	5000	96,153.85	4.08	20,400	392,307.69	296,153.84	4.26	21,300	409,615.38	313,461.50	17,307.69
1	***4800***	***92,307.69***	***4.08***	***19,584***	***376,538.46***	***284,230.77***	***4.26***	***20,448***	***393,076.92***	***300,769.20***	***16,538.46***

Discounted and marginal profit in Euro.

Summary and future:

This study summarizes the literature on GAM models for improving the accuracy of customer satisfaction prediction models or churn prediction models. We establish that it is beneficial to use the GA techniques to improve on the accuracy and thus improve on the profits.

However, to validate the modeling techniques externally, GAM can be applied to situations other than churn contexts. It may also be applied to classification problems other than binary. They may be related to customer relationship management or other fields. Also, this study only compares the logistic models with GAM, future research can be done where comparison is between models that can both fit non-linear data, like decision trees, neutral networks, random forest, etc.

Reviewed by: Nirav Thaker

Reference: Do we need hundreds of classifiers to solve real world classification problems by Manuel Fernandez-Delgado, Senen Barro and Eva Cernadas; http://www.jmlr.org/papers/v15/delgado14a.html

Objective:

g) Key problem: Identify the best classifiers by testing them on 121 datasets with 179 classifiers.
h) Key reason for the paper: To identify the best performing classifier overall.
i) Key objective: The objective is to achieve conclusions on classifier behavior, not depending on the data at hand. So, we perform the modeling on 121 data sets available on UCI.
j) How did the authors address the problem: They evaluated 179 classifiers arising from 17 families and tested them on 121 data sets. This would give conclusions that could be applicable to any real world problem.
k) What did they find: The models achieving maximum accuracy are RandomForest, SVM and LibSVM in the same order.
l) What does it imply: The difference between the accuracy of RandomForest and SVM is not significant but RF performs better overall.

Approach:

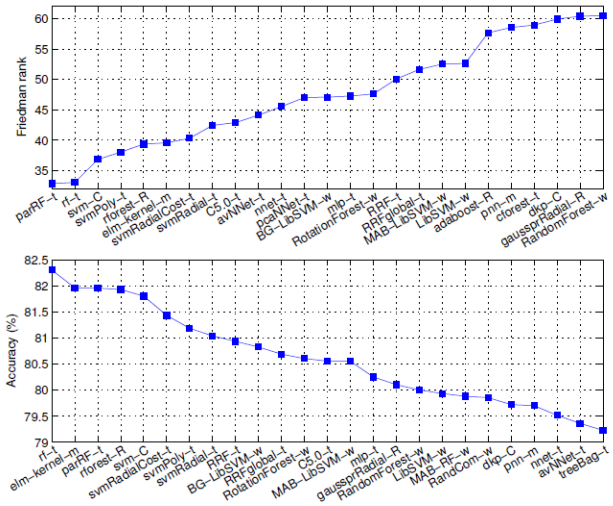
179 classifiers used are implemented in C/C++, R, Matlab, Weka.
179 classifiers with 121 datasets give a combination of 21,659 classifier-dataset. There were errors in many of the classifiers due to various reasons like collinearity, discrete inputs, patterns, etc. These errors are excluded to calculate the average accuracy. Cross validation is performed using the whole available data. Friedman ranking and Cohen K is calculated for each classifier.
However, this methodology may not be applied to large data sets.
In the end the results are those over 4 tests.

Results:

The top ranked classifiers are as follows:

Rank	Acc.	κ	Classifier	Rank	Acc.	κ	Classifier
32.9	82.0	63.5	parRF_t (RF)	67.3	77.7	55.6	pda_t (DA)
33.1	**82.3**	**63.6**	rf_t (RF)	67.6	78.7	55.2	elm_m (NNET)
36.8	81.8	62.2	svm_C (SVM)	67.6	77.8	54.2	SimpleLogistic_w (LMR)
38.0	81.2	60.1	svmPoly_t (SVM)	69.2	78.3	57.4	MAB_J48_w (BST)
39.4	81.9	62.5	rforest_R (RF)	69.8	78.8	56.7	BG_REPTree_w (BAG)
39.6	82.0	62.0	elm_kernel_m (NNET)	69.8	78.1	55.4	SMO_w (SVM)
40.3	81.4	61.1	svmRadialCost_t (SVM)	70.6	78.3	58.0	MLP_w (NNET)
42.5	81.0	60.0	svmRadial_t (SVM)	71.0	78.8	58.23	BG_RandomTree_w (BAG)
42.9	80.6	61.0	C5.0_t (BST)	71.0	77.1	55.1	mlm_R (GLM)
44.1	79.4	60.5	avNNet_t (NNET)	71.0	77.8	56.2	BG_J48_w (BAG)

The best ranked classifier is ParRF_t and the second is rf_t. This result is surprising considering that RandomForest is an old method compared to the various new ones and it still performs better.
The Friedman ranking and average accuracies are as follows:



Probability of achieving the Maximum accuracy is also calculated which shows th number of data sets in which it achieves the maximum accuracy. The results are as follows:

No.	Classifier	PAMA	No.	Classifier	PAMA
1	elm_kernel_m	13.2	11	mlp_t	5.0
2	svm_C	10.7	12	pnn_m	5.0
3	parRF_t	9.9	13	dkp_C	5.0
4	C5.0_t	9.1	14	LibSVM_w	5.0
5	adaboost_R	9.1	15	svmPoly_t	5.0
6	rforest_R	8.3	16	treebag_t	5.0
7	nnet_t	6.6	17	RRFglobal_t	5.0
8	svmRadialCost_t	6.6	18	svmlight_C	5.0
9	rf_t	5.8	19	Bagging_RandomForest_w	4.1
10	RRF_t	5.8	20	mda_t	4.1

No.	Classifier	P95	No.	Classifier	P95
1	parRF_t	71.1	11	elm_kernel_m	60.3
2	svm_C	70.2	12	MAB-LibSVM_w	60.3
3	rf_t	68.6	13	RandomForest_w	57.0
4	rforest_R	65.3	14	RRF_t	56.2
5	Bagging-LibSVM_w	63.6	15	pcaNNet_t	55.4
6	svmRadialCost_t	63.6	16	RotationForest_w	54.5
7	svmRadial_t	62.8	17	avNNet_t	53.7
8	svmPoly_t	62.8	18	nnet_t	53.7
9	LibSVM_w	62.0	19	RRFglobal_t	53.7
10	C5.0_t	61.2	20	mlp_t	52.1

No.	Classifier	PMA	No.	Classifier	PMA
1	parRF_t	94.1	11	RandomCommittee_w	91.4
2	rf_t	93.6	12	nnet_t	91.3
3	rforest_R	93.3	13	avNNet_t	91.1
4	C5.0_t	92.5	14	RRFglobal_t	91.0
5	RotationForest_w	92.5	15	knn_R	90.5
6	svm_C	92.3	16	Bagging-LibSVM_w	90.5
7	mlp_t	92.1	17	Bagging_REPTree_w	90.4
8	LibSVM_w	91.7	18	MAB_MLP_w	90.4
9	RRF_t	91.4	19	elm_m	90.3
10	dkp_C	91.4	20	rda_R	90.3

In the same way, we use Bagging, Boosting, GLM to rank the classifiers according to their accuracy and other parameters.

Summary:

This is a study of the behavior of 179 classifiers when tested for 121 data sets that are from the real world problems. The best results are achieved by using ParRF_t which provides 94.1% accuracy over all datasets and overcomes 90% accuracy in 102 out of 121 datasets.

Thus, ParRF_t may be used as a reference to compare the other classifiers with. Best performing classifiers are tuned in R which may be used for such problems.

Reviewed by: Sushant Chittoor Ravinder

Complete reference: Customer churn prediction using improved balanced random forests; Y Xie, X Li, EWT Ngai, W Ying; Expert systems with applications [0957-4174] Xie yr: 2009 vol: 36 iss: 3 pg: 5445 -5449

Objective:

a) What is the key problem the paper is addressing?
The key problem this paper is addressing is how to deal with imbalance in data distribution.
b) What is the key reason why addressing this problem is important?
In many real world applications like, for example, churn prediction, imbalance in data distribution is a very relevant and demanding problem.
c)What is the key objective this paper addressing?
In this paper, the authors propose a new method called Improved Balanced Random Forest (IBRF) method, obtained by combining Balanced Random Forests and Weighted Random Forests, to predict churn in the banking industry.
d)How are the authors addressing this problem?
According to the authors, this method helps address the problem of imbalance in data by altering the class distribution and by putting higher penalties on misclassification of the minority class.
e) What did they find?
They found out that the IBRF method produces a higher accuracy than other random forest algorithms such as balanced random forests and weighted random forests.
f) What does it imply?
This implies that IBRF is an effective method to gain higher accuracy when there is imbalance in data distribution.

Approach:

The authors combine the sampling techniques of balanced random forest and the cost-sensitive learning of weighted random forest to change the class distribution and penalize more heavily the misclassification of the minority class. To combine these two methods, they introduce two "interval variables" m and d, where m is the middle point of an interval and d is the length of the interval.
The algorithm used by the authors takes a training set as input and splits it into two subsets with one set containing all the positive samples while the other set having all the negative samples. For $t=1,2...n_{tree}$, a variable α is randomly generated within the interval $m - d/2$ and $m + d/2$. $n\alpha$ sample is drawn with replacement from the negative training dataset and $n(1-n\alpha)$ sample is drawn with replacement from the positive training dataset. An unpruned classification tree is grown.
All test samples are ordered by the negative score of each sample. The negative score of each sample is considered to be the total number of trees which predict the sample to be negative. The more trees predict the sample to be negative, the higher negative score the sample gets.

Results:

Table 2
Experimental results of different algorithms

Algorithm	ANN	DT	CWC-SVM	IBRF
Accuracy rate	78.1%	62.0%	87.2%	93.2%
Top-decile lift	2.6	3.2	3.5	7.1

ANN, artificial neural network; DT, decision tree; IBRF, improved balanced random forests.

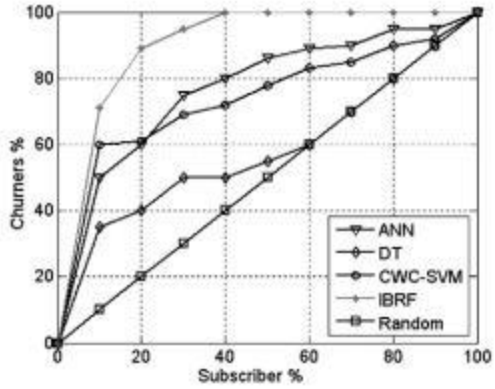


Fig. 2. Lift curve of different algorithms.eps.

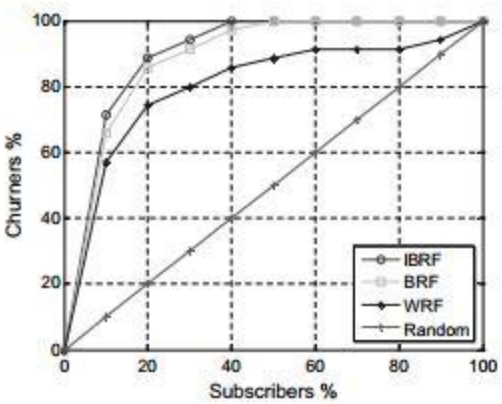


Fig. 3. Lift curve of different random forests algorithms.eps.

From the figures we can say that the IBRF is better classifier as its lift curve is higher than the other algorithms.

Summary:

The major takeaways from this work are as follows:

- The authors combine the sampling techniques of balanced random forest and the cost-sensitive learning of weighted random forest to change the class distribution and penalize more heavily the misclassification of the minority class.
- According to the study, the best features are iteratively learned by artificially making class priors equal, based on which best weak classifiers are derived.
- According to the study, IBRF produces higher accuracy than other random forests algorithms such as balanced random forests and weighted random forests

Also, IBRF employs internal variables to determine the distribution of samples. Although the results are found to be insensitive to the values of these variables, imposing some limitations on them in future experiments may enhance the predictive effectiveness of the method. In addition, there is further research potential in the inquiry into the cost-effectiveness of this method.

Reviewed by: Sushant Chittoor Ravinder

Reference: **Global Optimization Ensemble Model for Classification Methods**; Hina Anwar, Usman Qamar, Abdul Wahab Muzaffar Qureshi; Vol. 2014, pp. 313164. Date of Electronic Publication: 2014 Apr 27.

Objective:

a) What is the key problem the paper is addressing?

Some basic issues like bias-variance tradeoff, dimensionality of input space, and noise in the input data space, that affect the accuracy of a classifier, are dealt with in this paper.

b) What is the key reason why addressing this problem is important?

There is no generalized improvement method that can deal with the issues mentioned above and also improve the accuracy of a classifier.

c)What is the key objective this paper addressing?

This paper proposes a global optimization ensemble model for classification methods (GMC) that can improve the overall accuracy for supervised learning problems.

d)How are the authors addressing this problem?

The authors designed the GMC in layers with each layer solving one of the basic issues mentioned above.

e) What did they find?

The authors showed through experimentation that if the classifiers were enclosed in the GMC, the accuracy improved from 1% to 30% depending upon the algorithm complexity and its capability of handling bias and variance.

f) What does it imply?

This implies that the GMC used by the authors yielded better results than when the classifiers were used alone or in combination.

Approach:

The authors designed the global ensemble model by carrying out operations in different layers as shown in the figure below. In each of the layers, the following actions are executed.

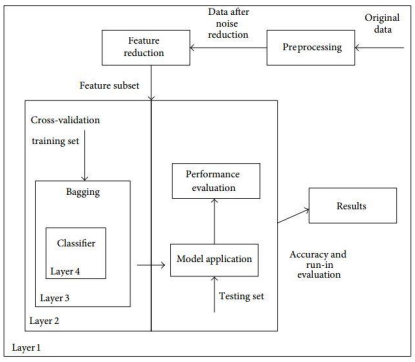


FIGURE 1: Design of global optimization ensemble model for classification methods (GMC).

- **Layer 1:** Genetic Algorithm is used for feature selection
- **Layer 2:** Partition of data is done using X- Fold Cross Validation
- **Layer 3:** Bagging is done to obtain the optimal bias-variance trade-off.
- **Layer 4:** Classifiers are placed in this layer.

This approach helped the authors achieve greater accuracies when used on various public datasets.

Results:

The authors used their model on 7 different datasets, details and results of which are given below.

Table 4: Data set details.

Data set	Number of cases	Number of attribute	Number of classes	Attribute characteristics	Missing values
Cancer dataset	699	9	2	Numeric	Yes
Diabetes dataset	768	9	2	Integer and real	No
Heart disease dataset	303	14	2	Categorical, integer, and real	Yes
Adult income dataset	1000	15	2	Integer and nominal	No
Wine dataset	178	13	3	Real and integer	No
Sonar dataset	208	61	2	Real and nominal	yes
Educational progress dataset	50	9	3	Nominal	No

Table 6: Results for cancer dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	66.81%	96.57%	29.76%
Decision tree	94.42%	96.71%	2.29%
ID3	66.52%	85.27%	18.52%
W-PART	94.71%	97.28%	2.57%
W-Prism	90.13%	96.28%	6.15%
W-J48	94.71%	96.71%	2%
W-AODE	97.00%	100%	3%
Logistic regression	95.01%	96.14%	1.13%

Table 7: Results for heart disease dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	50.82%	59.75%	8.93%
Decision tree	44.89%	59.43%	14.54%
ID3	47.52%	55.48%	8.24%
W-PART	50.52%	60.08%	9.56%
W-Prism	47.51%	56.09%	8.58%
W-AODE	55.47%	61.13%	5.66%
W-J48	49.87%	61.05%	11.18%
Rule induction	57.72%	59.76%	2.4%

Table 12: Results of diabetes dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	73.70%	77.48%	4%
Decision tree	74.0%	75.39%	1.39%
W-PART	73.83%	77.34%	3.51%
W-Prism	57.42%	67.97%	10.55%
W-J48	74.08%	77.22%	3.14%
W-AODE	66.54%	69.14%	2.6%
Logistic regression	76.00%	77.95%	1.95%

Table 8: Results of wine dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	70.75%	90.42%	19.67%
Decision tree	91.57%	95.49%	3.92%
W-PART	90.42%	96.67%	6.25%
W-Prism	52.32%	61.27%	8.95%
W-AODE	71.34%	75.26%	3.92%
W-J48	90.46%	96.63%	6.17%
Rule induction	86.37%	93.27%	6.9%

Table 9: Results of adult income dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	76.70%	83.20%	6.5%
Decision tree	80.00%	82.20%	2.20%
ID3	75.60%	78.60%	3%
W-PART	81.00%	83.50%	2.4%
W-Prism	81.30%	82.20%	1.1%
W-AODE	80.80%	82.60%	1.8%
W-J48	81.50%	83.00%	1.5%
Random forest	76.10%	77.30%	1.2%
Logistic regression	79.00%	80.00%	1%

Table 10: Results of sonar dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	69.71%	74.57%	4.86%
Decision Tree	73.57%	83.67%	10.1%
W-PART	75.48%	83.17%	7.69%
W-Prism	48.02%	63.38%	15.36%
W-J48	70.24%	82.21%	11.97%
Rule induction	71.66%	76.48%	4.82%
Random forest	68.26%	75.36%	7.1%
Logistic regression	74.55%	80.29%	5.74%

Table 11: Results of educational dataset: comparison of optimized classification accuracy using GMC model with simple classification using different classifiers.

Algorithm	Classification accuracy	Optimized classification accuracy	Improvement %
K-NN	46%	54%	8%
Decision Tree	42%	56%	14%
ID3	20%	44%	24%
W-PART	32%	54%	22%
W-Prism	24%	50%	26%
W-J48	44%	58%	14%
W-AODE	46%	56%	10%
SVM	60%	76%	16%
Random forest	48%	58%	12%
Rule induction	44%	54%	10%

In this study, the experimental results on various public datasets showed that the proposed model improved the accuracy of the classification models from 1% to 30% depending upon the algorithm complexity.

Summary:

The major takeaways from this work are as follows:

- Some basic issues of supervised learning problems like dimensionality reduction, bias-variance tradeoff and noise can be countered by using the concept of ensemble models to design an optimized global ensemble model(GMC)
- The model was designed in layers with each layer addressing one of the issues.

Reviewed by: Muzammil Bashir

Reference: A Comparative Assessment of the Performance of Ensemble Learning in Customer Churn Prediction by Hossein Abbasimehr, Mostafa Setak and Muhammad Tarokh, The International Arab Journal of Information Technology, Vol. 11, No. 6, November 2014.

Objective

a) What is the key problem the paper is addressing?

The tendency of a customer to leave a company (customer churn) is of major concern for any business. This paper uses data analytics to find the best prediction model for customer churn by evaluating performance indicators (AUC, sensitivity and specificity) for different analytics ensemble models.

b) What is the key reason why addressing this problem is important?

It has been shown that ensemble methods (methods that use a combination of classifier models) shows a better performance over individual models in predicting customer churn. Of the four methods evaluated in the paper, two haven't been evaluated for churn prediction previously.

c) What is the key objective this paper is addressing?

The objective of this paper is to find the prediction model that gives the best result in classifying customers with a high probability to leave. This classification will help in directing marketing resources towards customers more prone to leaving.

d) How are the authors addressing this problem?

The authors addressed this problem by evaluating the prediction performance of four ensemble methods Bagging, Boosting, Stacking and Voting based on base learners Decision tree (DT), Artificial neural network (ANN), support vector machine (SVM) and reduced incremental pruning to produce error reduction (RIPPER).

e) What did they find?

This paper states that the Boosting based on C4.5 decision tree and reduced incremental pruning to produce error reduction (RIPPER) shows the best results for churn prediction tasks. Basic sampling technique for balancing datasets showed comparatively better result.

f) What does it imply?

It implies that ensemble techniques have an advantage over basic learning algorithms when it comes to binary classification in customer churn context. These improved prediction methods can help in retaining customers which is vital for any business.

Approach

The authors address the problem of predicting customer churn by reviewing literature and finding that ensemble methods have shown improvement over base learning techniques. In addition to Boosting and bagging that have been more commonly studied for this type of predictions, authors additionally evaluated stacking and voting techniques.

In the feature selection step, PART (partial decision tree) algorithm was applied to reduce the number of features. The data used in this case was imbalanced due to skewed distribution among the classes. Both basic and advance sampling techniques were applied to reduce the imbalance among the classes. Oversampling and Synthetic minority over-sampling technique (SMOTE) were applied on all four base

learner models, resulting in a total of 8 base learner models. Base learner algorithms (C4.5, ANN, SVM, RIPPER) were applied on the balanced dataset, best parameters of each of these algorithms was obtained. Performance criteria for selection of the best prediction model was based on area under curve AUC after application of ROC analysis.

The best base classifiers were then further augmented through use of ensemble techniques of Bagging, Boosting, Staking and voting. All 16 resulting models were evaluated based on the performance criteria and reported.

Results

Table 3. Ensemble learning results.

Ensemble method	Base learner	AUC	Sens	Specs
Bagging	C4.5	0.999	1	0.984
	RIPPER	0.997	0.942	0.98
	ANN	0.941	0.862	0.972
	SVM	0.987	0.987	0.96
Boosting	C4.5	1	1	0.986
	RIPPER	1	1	0.988
	ANN	0.966	0.875	0.984
	SVM	0.966	1	0.962
Staking	C4.5	0.992	0.973	0.962
	RIPPER	0.988	0.978	0.99
	ANN	0.998	0.978	0.991
	SVM	0.978	0.982	0.974
Voting	ALL base learners	0.998	0.978	0.984

Table 2. Base learners results.

Technique	Sampling technique	AUC	Sens	Spec
C4.5	Oversampling	0.983	1	0.958
C4.5	SMOTE	0.886	0.786	0.963
Ripper	Oversampling	97.7	0.977	0.977
Ripper	SMOTE	88.3	0.773	0.985
ANN	Oversampling	0.94	0.87	0.981
ANN	SMOTE	0.92	0.844	0.936
SVM	Oversampling	0.969	0.978	0.96
SVM	SMOTE	0.876	0.835	0.917

It can be inferred from the above Table 2. That the basic sampling technique of oversampling provides much better results as opposed to the advanced SMOTE. It is also evident that the Boosting ensemble technique based on C4.5 and RIPPER algorithms provide the best result for this case.

Summary

In order to improve performance of prediction model, ensemble technique can be applied. The paper investigates two types of sampling & four types of ensemble techniques on the base learning algorithms. It was noted that oversampling improves the prediction performance, and that the application of ensemble significantly improved the predictive power of the base learners. Boosting came out to be the best candidate for churn prediction tasks. Since our project is related to customer churn, this paper will help in applying the above mentioned ensemble method.

Reviewed By: Muzammil Bashir

Reference: An Introduction to variable and Feature selection. Isabelle Guyon, Andre Elisseeff. Journal of Machine Learning Research 3 (2003) 1157 – 1182.

Reviewed By: Muzzamil Bashir

Objective

a) What is the key problem the paper is addressing?

The paper discusses the problem of dealing with thousands of variables in the dataset. Selection of variables that would improve the prediction and computational performance of the chosen classification algorithm are extensively discussed.

b) What is the key reason why addressing this problem is important?

The rapid increase in the domain of data analytics have resulted in problems with a much greater number of predictors, most of which could be irrelevant or redundant. Performing analysis on huge set of predictors is not only computationally intense but also complicates the underlying processes that resulted in the data.

c) What is the key objective this paper is addressing?

The paper investigates various variable selection techniques and their limitation across a wide domain of study. It focuses on selecting subsets of features as opposed to individually ranking variables.

d) How are the authors addressing this problem?

The authors are addressing the problem through discussing the available techniques of variable reduction, their limitation and comparative advantages. Various methods of variable selection including ranking, feature construction and subset selection are discussed in detail.

e) What did they find?

It was concluded that sophisticated subset selection techniques of wrapper or embedded methods have prediction advantage over ranking method, but these advantages might not be significant for each of the domains studied and varies through different type of datasets.

f) What does it imply?

It implies that a unifying variable selection method couldn't be developed due to the diverse nature of problems studied. A recommendation is made to use a linear prediction model that employs filtration as a preprocessing step followed by wrapper / embedded techniques. The final assessment could be made based on the best reported results.

Approach

The paper covers a range of topics on variable selection in data analytics. The study uses datasets from a wide variety of study domains, as shown in the attached figure. It starts with discussion of ranking techniques that rank individual variables through correlation coefficients and their limitations through use of constructed examples. Filters as an independent pre-processing tool for feature selection is discussed.

Variable subset selection is then elaborated as a solution to the limitations of the ranking techniques. In this section wrapper (scoring subsets of variable according to their predictive power) and embedded methods (variable selection in training) are discussed. These methods employ machine learning algorithms and are extensively used in feature selection, however they are computationally intense.

Data set	Description	patterns	variables	classes	References
Linear[a,b]	Artificial linear	10-1200	100-240	reg-2	SWBe
Multi-cluster[c]	Artificial non-linear	1000-1300	100-500	2	PS
QSAR[d]	Chemistry	30-300	500-700	reg	Bt
UCI[e]	ML repository	8-60	500-16000	2-30	ReBnToPC
LVQ-PAK[f]	Phoneme data	1900	20	20	T
Raetch bench.[g]	UCI/Delve/Statlog	200-7000	8-20	2	Ra
Microarray[a]	Cancer classif.	6-100	2000-4000	2	WRa
Microarray[a]	Gene classification	200	80	5	W
Aston Univ[h]	Pipeline transport	1000	12	3	T
NIPS 2000[i]	Unlabeled data	200-400	5-800	reg	Ri
20 Newsgroup[j,o]	News postings	20000	300-15000	2-20	GBkD
Text filtering[k]	TREC/OSHUMED	200-2500	3000-30000	6-17	F
IR datasets[l]	MED/CRAN/CISI	1000	5000	30-225	G
Reuters-21578[m,o]	newswire docs.	21578	300-15000	114	BkF
Open Dir. Proj.[n]	Web directory	5000	14500	50	D

Table 1: **Publicly available data sets used in the special issue.** Approximate numbers or ranges of patterns, variables, and classes effectively used are provided. The "classes" column indicates "reg" for regression problems, or the number of queries for Information Retrieval (IR) problems.

Transforming original variables into dimensionally reduced set of features is called feature construction. The paper discusses this aspect of dimensional reduction and its applicability. The paper then discusses approach for cross validation. Non parametric variable selection can be applied by introduction of fake variables in the dataset, these variables can provide elimination criteria for the variables that are comparatively insignificant to the fake variables. The paper concludes with some of the problems that the author encountered during the course of his cross domain study.

Results

Discussing limitation of ranking techniques (removing individual insignificant variables), the author states through relevant examples that there could be variables that individually are useless but can result in prediction improvement when used with other variables.

Filters are faster than wrappers and can be employed as a preprocessing step to reduce dimensionality and overcome fitting. Wrapper are simple and uses the knowledge from machine learning for variable selection. Wrapper can be computationally intense, however greedy search strategies of forward selection and backward elimination are computationally advantageous and avoid overfitting problems. Embedded methods like decision trees and CART are comparatively faster and have built in capability to variable selection by avoiding splitting of data into training and validation. When using fake variables, a halting criteria can be applied in forward selection.

Summary

The paper is an effort to provide a benchmark for variable selection by discussing all the relevant techniques available. It notes that the advantage of using wrapper or embedded methods does exist but it is not universal across different domains. Due to this, the paper is unable to establish a unifying criterion for variable selection and recommends the use of domain based criterion by evaluating both correlation coefficient and nested subset selection. A recommendation is made to use a linear prediction model that employs filtration as a preprocessing step followed by wrapper / embedded techniques.

This paper (8563 citation) is an in-depth critical review of feature selection and provides discussion on each of the variable selection methods. **These discussion would be insightful in variable selection of our intended project.**