2009 21st IEEE International Conference on Tools with Artificial Intelligence

A Study on the Relationships of Classifier Performance Metrics

Naeem Seliya *
Taghi M. Khoshgoftaar †
Jason Van Hulse ‡

Abstract

There is no general consensus on which classifier performance metrics are better to use as compared to others. While some studies investigate a handful of such metrics in a comparative fashion, an evaluation of specific relationships among a large set of commonly-used performance metrics is much needed in the data mining and machine learning community. This study provides a unique insight into the underlying relationships among classifier performance metrics. We do so with a large case study involving 35 datasets from various domains and the C4.5 decision tree algorithm. A common property of the 35 datasets is that they suffer from the class imbalance problem. Our approach is based on applying factor analysis to the classifier performance space which is characterized by 22 performance metrics. It is shown that such a large number of performance metrics can be grouped into two-to-four relationship-based groups extracted by factor analysis. This work is a step in the direction of providing the analyst with an improved understanding about the different relationships and groupings among the performance metrics, thus facilitating the selection of performance metrics that capture relatively independent aspects of a classifier's performance.

Keywords: binary classification; performance metrics; factor analysis; metrics relationship.

1 Introduction

A classifier is evaluated based on performance metrics computed after the model-training process. While various classifiers are available for machine learning purposes [11, 12, 14], there is no general consensus among practitioners regarding which performance metrics to use for evaluating a classifier's performance [8, 17]. It is not uncommon that a given classifier performs very well when evaluated with respect to a certain performance metric, while the same classifier performs poorly with respect to another performance metric. How does a practitioner know which performance metrics to use?

Generally, the analyst will choose one or more performance metrics for classifier evaluation because they are commonly used in the community. Examples of such metrics include classification accuracy, area under the ROC (receiver operating characteristic) curve, and F-measure [4, 8, 17]. One can find several studies that build classification models and make conclusions based on the selected performance metrics. Such an approach can lead to redundancy in performance space evaluation, ignoring some performance aspects, or make it difficult to obtain a clear winner when competing performance metrics provide conflicting conclusions.

Intuitively speaking, a classifier should be evaluated with respect to a set of performance metrics that individually capture a unique aspect of the classifier performance space. This becomes more important since most performance metrics are based on the four values (for a binary classifier) of the confusion matrix, i.e., number of true positives, number of true negatives, number of false positives, and number of false negatives. Consequently, it is logical to think that some of the derived performance metrics are likely to be closely related.

This study provides a direct insight into the relationships among various commonly used classifier performance metrics. To our knowledge, similar studies are absent in the current literature with the exception of a few which evaluate the relationship between a small set of performance metrics. Davis and Goadrich [4] investigated the relationship between precision-recall curves and ROC curves and found that an integral association exists between precision-recall space and ROC space. Alaiz-Rodriguez et al. [1] conclude that classifier evaluation should be done on an exploratory basis by considering several performance metrics and multiple datasets, rather than choosing a few select performance metrics and a limited number of datasets. Caruana

*Computer and Information Science, University of Michigan – Dearborn, 4901 Evergreen Rd., Dearborn, MI 48128. Email: nseliya@umich.edu.

†Computer Science and Engineering, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL 33431. Email: taghi@cse.fau.edu.

‡Computer Science and Engineering, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL 33431. Email: jvanhulse@gmail.com.

1082-3409/09 $26.00 © 2009 IEEE
DOI 10.1109/ICTAI.2009.25

and Niculescu-Mizil [3] investigate multidimensional scaling as a way to visualize the classifier performance space with respect to a set of performance metrics.

In this work, a large case study involving 35 datasets from various application domains and the C4.5 decision tree classifier presents a unique perspective on relationships among 22 performance metrics. We selected the C4.5 classifier because it is one of the most frequently used learners. A unique property common among all of the case study datasets is that they suffer from the class imbalance problem, i.e., the minority class size is disproportionately smaller compared to the majority class size. The underlying strategy of our approach is the use of the statistical method of factor analysis to obtain a small number of factors to group a large set of performance metrics into relationship-based clusters. To our knowledge, this is the first study to investigate the relationships between several classifier performance metrics using factor analysis.

This study will provide practitioners and researchers with a better understanding on selecting performance metrics for classifier evaluation. The knowledge of the relationships and groupings among the commonly-used performance metrics will allow them to select contrasting performance metrics, thereby evaluating different aspects of the selected classification model. Our aim is not to provide a recommendation on performance metrics, but instead, examine what relationships exist among the performance metrics, and why such relationships might exist.

The remainder of the paper is structured as follows: Section 2 summarizes the 22 performance metrics considered in our evaluation; Section 3 details the basic background on factor analysis; Section 4 discusses our case study including selected classifiers, datasets, and empirical results; and, Section 5 summarizes our work and provides suggestions for future work.

2 Classifier Performance Metrics

In a binary classification problem, a two-by-two confusion matrix depicts the numbers of instances predicted by each of the four possible outcomes: number of true positives (*#TP*), number of true negatives (*#TN*), number of false positives (*#FP*), and number of false negatives (*#FN*). Most classifier performance metrics are derived from the four values.

A given classification algorithm may use a classification strategy where a parameter, called the decision threshold t ($0 \leq t \leq 1$), is used to decide the class membership of a given instance. The default decision threshold is 0.5 and implies that the class with the higher posterior probability is used as the final prediction. The decision threshold can be changed to account for class imbalance in the dataset or for unequal costs of misclassification. More specifically, lowering the decision threshold generally causes an increase in the number of *FP* errors while reducing the number of *FN* errors.

The 22 classifier performance metrics used in our study include: *AUC*, *BRI*, *LGS*, *KSS*, *DVG*, *BFM*, *BGM*, *DFPR*, *DFNR*, *DPPV*, *DNPV*, *DACR*, *DFM*, *DGM*, *KFPR*, *KFNR*, *KPPV*, *KNPV*, *KACR*, *KFM*, *KGM*, and *PRC*. These metrics (including their abbreviations) are explained in the subsequent subsections.

2.1 Accuracy and Predictive Values

Let N be the total number of instances in the dataset, N_{c_1} be the total number of positive instances and N_{c_0} be the total number of negative instances in the dataset. Let t denote the decision threshold at which a given classification-based performance metric was obtained. The following equations define the performance metrics related to accuracy and predictive values:

$$ACR(t) = \frac{\#TP(t) + \#TN(t)}{N} \quad (1)$$

$$MCR(t) = \frac{\#FP(t) + \#FN(t)}{N} \quad (2)$$

$$TPR(t) = \frac{\#TP(t)}{N_{c_1}} \quad (3)$$

$$TNR(t) = \frac{\#TN(t)}{N_{c_0}} \quad (4)$$

$$FPR(t) = \frac{\#FP(t)}{N_{c_0}} \quad (5)$$

$$FNR(t) = \frac{\#FN(t)}{N_{c_1}} \quad (6)$$

$$PPV(t) = \frac{\#TP(t)}{\#TP(t) + \#FP(t)} \quad (7)$$

$$NPV(t) = \frac{\#TN(t)}{\#TN(t) + \#FN(t)} \quad (8)$$

$$DACR = \frac{\#TP(0.5) + \#TN(0.5)}{N} \quad (9)$$

$$DPPV = \frac{\#TP(0.5)}{\#TP(0.5) + \#FP(0.5)} \quad (10)$$

$$DNPV = \frac{\#TN(0.5)}{\#TN(0.5) + \#FN(0.5)} \quad (11)$$

$$DFPR = \frac{\#FP(0.5)}{N_{c_0}} \quad (12)$$

$$DFNR = \frac{\#FN(0.5)}{N_{c_1}} \quad (13)$$

where $ACR(t)$ and $MCR(t)$ represent the classification accuracy and misclassification rates at decision threshold t, respectively; $TPR(t)$, $TNR(t)$, $FPR(t)$, and $FNR(t)$ represent the true positive rate, true negative rate, false positive rate, and false negative rate at decision threshold t,

respectively; $PPV(t)$ and $NPV(t)$ represent the positive predictive value (also known as *Precision*) and negative predictive value at decision threshold t, respectively; and $DACR$, $DPPV$, $DNPV$, $DFPR$, and $DFNR$ represent the accuracy, positive predictive value, negative predictive value, false positive rate, and false negative rate at the default decision threshold, respectively.

2.2 F-measure

The F-measure (*FM*) is a single value metric based on two parameters [20], *Recall* and *Precision*, as shown by Equation 14. Precision (positive predictive value) is defined earlier, while $Recall = \frac{\#TP(t)}{\#TP(t)+\#FN(t)}$. The Default F-measure (*DFM*) corresponds to a decision threshold value of 0.5, while the Best F-measure (*BFM*) corresponds to the t value that maximizes it and where $0 \leq t \leq 1$. The F-measure ranges between 0 and 1 such that a perfect classifier yields an F-measure of 1, i.e., when both recall and precision equal 1. A value of 1 is used for β in our study.

$$FM = \frac{(1+\beta^2) \times Recall \times Precision}{Recall + Precision} \quad (14)$$

$$DFM = FM(0.5) \quad (15)$$

$$BFM = \max_{t \in [0,1]} \text{FM}(t) \quad (16)$$

2.3 Geometric Mean

The Geometric Mean (*GM*) is a single-value performance measure that ranges from 0 to 1 where a perfect classifier provides a value of 1. It is a useful performance measure since it is inclined to maximize the true positive rate and the true negative rate while keeping them relatively balanced. Such error rates are often preferred, depending on the misclassification costs and the application domain. The threshold t equals 0.5 for the Default Geometric Mean (*DGM*). The Best Geometric Mean (*BGM*) is the maximum Geometric Mean value that is obtained for $0 \leq t \leq 1$.

$$GM(t) = \sqrt{TPR(t) \times TNR(t)} \quad (17)$$

$$DGM = GM(0.5) \quad (18)$$

$$BGM = \max_{t \in [0,1]} \text{GM}(t) \quad (19)$$

2.4 Area Under the ROC Curve

The area under the ROC (Receiver Operating Characteristic) curve (i.e., *AUC*) is a single-value measurement, with its value ranging from 0 to 1. The ROC curve is used to characterize the trade-off between hit (true positive) rate and false alarm (false positive) rate [4, 5]. It depicts the performance of a classifier without taking class distribution or error costs into consideration. A classifier that provides a large area under the curve is generally preferable over a classifier with a smaller area under the curve.

2.5 Area Under the Precision-Recall Curve

The area under the precision-recall curve (*PRC*) is a single-value measure, with values ranging from 0 to 1. The PRC diagram depicts the trade off between *Recall* and *Precision* [4]. A perfect classifier results in an area under the PRC of 1.

2.6 Logarithmic Score

The logarithmic score (LGS) is a single value that ranges from 0 to +∞, and where the value provided by a perfect classifier is 0. The logarithmic score is based on the cross-entropy ($\overline{CE}$) which is defined as [19]:

$$\overline{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{1} p(c = j \mid x^i) \times \ln(\hat{p}(c = j \mid x^i)) \quad (20)$$

where $x^i = (x_1^i, ..., x_n^i)^T \in \mathrm{R}^n$ denotes the input vectors of the dataset, $c \in \{0, 1\}$ denotes the class label of the current instance, $j \in \{0, 1\}$ denotes the possible class labels, $\hat{p}$ denotes the predicted class membership probabilities, and N denotes the number of instances in the dataset [19]. Since the true posterior class membership probabilities are unknown, $p(c = j \mid x^i)$ is replaced with $\delta(j, c_i)$, which is used to estimate the posterior class membership probabilities [19]. $\delta(j, c_i) = 1$ if $j = c_i$, otherwise, $\delta(j, c_i) = 0$. This replacement yields the logarithmic score.

2.7 Brier Inaccuracy

The Brier Inaccuracy (BRI) is a single value metric that ranges from 0 to 2, and where a perfect classifier provides a value of 0. The Brier Inaccuracy is defined as:

$$\overline{BS} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{1} (\hat{p}(c = j \mid x^i) - p(c = j \mid x^i))^2 \quad (21)$$

where $x^i = (x_1^i, ..., x_n^i)^T \in \mathrm{R}^n$ denotes the input vectors of the dataset, $c \in \{0, 1\}$ denotes the class label, $j \in \{0, 1\}$ denotes the possible class labels, $\hat{p}$ denotes the predicted class membership probabilities, and N denotes the number of instances in the dataset [19]. Since the true posterior class membership probabilities are not available, $p(c = j \mid x^i)$ is replaced with $\delta(j, c_i)$ in Equation 21 to obtain the Brier Inaccuracy.

2.8 Divergence

The Divergence (DVG) is a single value metric that ranges from 0 to +∞ [7, 16]. Hand and Henley [6] explain that the Divergence statistic is a separability measure. It is defined as follows:

$$\delta = \frac{(E[p(x) \mid c_1] - E[p(x) \mid c_0])^2}{.5\,(\text{var}[p(x) \mid c_1] + \text{var}[p(x) \mid c_0])}, \quad (22)$$

where $E[p(x) \mid c_1]$, $E[p(x) \mid c_0]$, var$[p(x) \mid c_1]$, and var$[p(x) \mid c_0]$ are the means and variances of the posterior probabilities $p(x)$ of positive and negative class examples, respectively. δ is undefined if var$[p(x) \mid c_1]$ and var$[p(x) \mid c_0]$ are 0, which rarely occurs in practice. Otherwise, $\delta \in [0, \infty]$, and a small value for δ implies that the classifier shows little separation between the class-conditional posterior probabilities, and hence the classifier has weak predictive power. An increasing δ indicates stronger predictive performance of a classifier.

2.9 Kolmogorov-Smirnov Statistic

The Kolmogorov-Smirnov Statistic (*KSS*) provides a single value that ranges from 0 to 1. It measures the maximum difference between the cumulative distribution functions of the predicted probabilities of examples in each class [10]. The distribution function $F_{c_i}(t)$ for a class c_i is estimated by the proportion of examples x from class c_i that have a posterior probability of class c_1 membership $p(c_1 \mid x) \leq$ decision threshold t, $0 \leq t \leq 1$. Given the distribution function

$$F_{c_i}(t) = \frac{\text{\# class } c_i \text{ instances with } p(c_1 \mid x) \leq t}{\text{\# class } c_i \text{ instances}},$$

KSS is computed as

$$KSS = \max_{t \in [0,1]} |Fc_1(t) - Fc_0(t)| \quad (23)$$

With respect to KSS, a classifier provides the best performance at a specific threshold t when the distance between the two distribution functions is maximized. The larger the KSS, the better the classifier is able to separate the two classes. A perfect classifier obtains a KSS of 1 and a classifier that is completely unable to separate the two classes results in a KSS of 0. In relation to the *KSS* metric, we collected (an additional) seven previously described performance metrics at the decision threshold where the *KSS* metric is maximized. In other words, the threshold that maximizes the KSS value is computed, the confusion matrix (i.e., #TP, #FP, #TN, #FN) is determined at this threshold, and then the following *KSS*-based metrics, abbreviated with a 'K' as the first letter, are calculated: KFPR (false positive rate), KFNR (false negative rate), KPPV (positive predictive value), KNPV (negative predictive value), KACR (accuracy), KFM (F-measure), and KGM (geometric mean).

3 Factor Analysis

The objective of factor analysis is to group the variables based on some other unobservable variables called factors. The number of selected factors is significantly smaller than the number of variables, and the factors represent the most simple structure that adequately explains the original dataset. The factor model argues that, if one was to group the variables so that all variables within a given group are highly related but have relatively small relationships with variables in other groups, then in each group there is probably an underlying factor that is responsible for the correlation [13].

There are two approaches to factor analysis: principal components analysis and common factor analysis, which are conceptually different. The use of principal components analysis is appropriate for the reduction of a set of related variables to a smaller set so that the extracted variables are not correlated while still adequately representing the original set. Common factor analysis is used to determine the underlying factors which influence a given set of variables, i.e., the goal is to express the original variables as linear combinations of a small number of hidden, non-measurable factors. We used common factor analysis in this study.

Suppose an observable $(p \times 1)$ random vector $\mathbf{X} = (X_1, \ldots, X_p)$ has mean vector μ and covariance matrix Σ. The factor model assumes that $\mathbf{X}$ can be written as [9]:

$$\mathbf{X} = \mu + \mathbf{LF} + \epsilon$$

where $\mathbf{L}$ is the $(p \times m)$ matrix of factor loadings, $\mathbf{F}$ is the $(m \times 1)$ vector of factors, and ϵ is the $(p \times 1)$ vector of errors. The number of factors is generally assumed to be much smaller than the number of attributes $(m < p)$, as the objective of factor analysis is to uncover the small number of factors which encapsulate the hidden, underlying relationships between attributes. The variable X_i is a linear combination of the factor loadings times the hidden factors plus some error. The error can also be thought of as the additional source of information that is specific to variable X_i.

It is common to place additional assumptions on $\mathbf{F}$ and ϵ (see [9]) such as $Cov(\mathbf{F}) = I$ (I is the $m \times m$ identity matrix with 1 on the diagonal and 0 off-diagonal), and it can be shown that $Cov(\mathbf{X}) = \mathbf{LL}^T + Cov(\epsilon)$. This implies that the variance of an attribute X_i is $\sum_{j=1}^{m} \ell_{ij}^2 + Var(\epsilon_i)$. In other words, we can measure how much of the variance of each attribute is accounted for by the common factors.

Table 1 shows how five variables are grouped based upon factor analysis with three selected underlying factors. The values shown in the rotated factor pattern table are called factor loadings and they range between -1 and 1. The factor loadings indicate how strong a certain variable is influenced

by the underlying factor. variable1 and variable3 are associated with Factor 1, variable2 and variable5 are associated with Factor 2, and variable4 is associated with only Factor 3. The factor loadings account for 81.41% of the variance of variable1 ($0.9^2 + 0.05^2 + 0.04^2$).

A factor that shows high factor loadings for all variables is called a general factor. A factor is called a common factor if at least two factor loadings are significantly different than 0. A factor that represents only one variable is called a unique factor. The sample rotated factor pattern in Table 1 does not contain any general factors. Factor 1 and Factor 2 are common factors while Factor 3 is a unique factor. The respective Variable-Factor associations are shown in bold face in Table 1. Note that the variables that are associated with the same factor have a high correlation coefficient in the corresponding correlation matrix, and variables that are associated with different factors have lower correlation coefficients in the correlation matrix. The rotated factor pattern provides a more structured view on the variable groupings while the correlation matrix provides more detail about the relationship between certain variables.

Factor analysis also provides a table of eigenvalues of the correlation matrix and a rotated factor pattern. The table of eigenvalues indicates how many underlying factors account for a certain amount of the total variance in the dataset. The number of factors equals the number of variables; however, only a few factors account for the vast majority of the overall variance in most real-world datasets. A threshold, called the mineigen criterion (minimum eigenvalue), is used to decide how many factors are used for the factor model. A value of 1 for the mineigen criterion is commonly used and often results in the coverage of the majority of the variance. There are other methods such as extracting as many factors as possible to explain, for example, at least 90% of the variance. In addition, values other than 1 can be used for the mineigen criterion. Factor analysis models generated in our experiments were performed using SAS [15]. The *varimax* option was used to rotate the factor patterns to allow for easier interpretability.

Table 1. Rotated Factor Pattern Example

Variable	Factor 1	Factor 2	Factor 3
variable1	**0.90**	0.05	0.04
variable3	**0.85**	0.10	0.00
variable2	0.01	**0.85**	0.35
variable5	0.10	**0.80**	0.00
variable4	0.17	0.35	**0.65**

4 Empirical Case Study

4.1 Classifiers and Datasets

The 35 datasets (shown in Table 2) that were used involve binary classification problems from various application domains such as satellite image evaluation, consumer credit research, software engineering, statistical medical research, character recognition, etc. Some of these datasets were obtained from the UCI Machine Learning Repository [2]. The datasets represent real-world problems, and involve challenging issues such as class imbalance [18].

The datasets vary in terms of the relative proportions of the two classes in a given dataset. The datasets also differ with regards to the number of attributes and the total number of instances – some datasets contain thousands of instances while others contain only a few hundred instances, and some datasets include less than 10 attributes while several others contain more than 30 attributes. We note that all datasets have, or were transformed to have, a binary class attribute since we only consider binary classification problems in this work.

The classification algorithms used in our study include two versions of the C4.5 decision tree algorithm available in the WEKA [20] data mining tool. The C4.5D version uses the default settings in WEKA, while the C4.5N version represents the algorithm with Laplace smoothing and without any pruning. A complete optimization of the C4.5 parameters is out of scope for this paper.

4.2 Results and Analysis

The 22 performance metrics were collected from each of the two classifiers (C4.5N and C4.5D) trained on each of the 35 datasets. 10-fold cross validation is used during training and the model building process was repeated 10 times (runs) to account for any bias during the formation of the cross validation data splits. Thus 7,000 classification models were built, i.e. 35 datasets $\times$ 2 classifiers $\times$ 10 runs $\times$ 10 folds. For a given classifier, the results (performance metrics) of the 10 folds were consolidated into one for each run, resulting into a total of 350 instances for a given performance metric. These instances were then used as the basis for factor analysis (for the given classifier) which provided a table of Eigenvalues computed from the correlation matrix.

The average performance metric values for both classifiers are shown in Table 3. A general observation is that the C4.5N classifier performs better than the C4.5D classifier. Note that the latter is based on using the default parameter settings for the C4.5 decision tree algorithm in WEKA, while C4.5N is based on selecting custom values for the pruning parameter. While a comparison of the two classifiers is not within the scope of this study, we present their performance metric values for completeness purposes.

Table 2. Case Study Datasets

Dataset	Instances	Minority %	Attributes
SP3	3533	1.33	43
SP4	3983	2.31	43
mammography	11207	2.32	7
nursery-3	12964	2.53	9
solar-flare-f	1390	3.67	13
letter-a	19975	3.95	17
car-3	1729	3.99	7
SP2	3979	4.75	43
cccs-12	282	5.67	9
SP1	3646	6.28	43
pc1	1106	6.87	16
mw1	403	7.69	16
glass-3	214	7.94	10
kc3	458	9.39	16
cm1	505	9.50	16
cccs-8	282	9.57	9
pendigits-5	10989	9.60	17
satimage-4	6434	9.73	37
optdigits-8	5619	9.86	65
e-coli-4	336	10.42	8
segment-5	2309	14.29	20
kc1	2108	15.42	16
jm1	8851	19.06	16
letter-vowel	20000	19.39	17
cccs-4	282	19.50	9
kc2	520	20.38	16
Contra-2	1473	22.61	10
SpliceJunc2	3189	24.08	61
vehicle-1	846	25.06	19
haberman	306	26.47	4
yeast-2	1484	28.91	9
phoneme	5404	29.35	6
cccs-2	282	29.43	9
german-credit	1000	30.00	21
pima-diabetes	768	34.90	9

Table 3. Average Performance Metric Values

#	Metric	C4.5N	C4.5D
1	AUC	0.8372	0.7391
2	BRI	0.1645	0.1715
3	LGS	0.2688	1.4051
4	KSS	0.6030	0.4839
5	DVG	6.0659	4.4414
6	BFM	0.5740	0.5304
7	BGM	0.7979	0.7229
8	DFPR	0.0663	0.0499
9	DFNR	0.4841	0.5409
10	DPPV	0.5466	0.5554
11	DNPV	0.9249	0.9221
12	DACR	0.8875	0.8958
13	DFM	0.5253	0.4899
14	DGM	0.6602	0.5989
15	KFPR	0.2009	0.1681
16	KFNR	0.1961	0.3530
17	KPPV	0.4335	0.4541
18	KNPV	0.9547	0.9343
19	KACR	0.8039	0.8112
20	KFM	0.5298	0.5010
21	KGM	0.7963	0.7060
22	PRC	0.5361	0.4618

Table 4. C4.5N: Rotated Factor Pattern

	Factor 1	Factor 2
KPPV	**0.9900**	0.0828
KFM	**0.9834**	0.1265
DFM	**0.9788**	0.1086
BFM	**0.9773**	-0.0487
PRC	**0.9650**	0.1351
DGM	**0.9586**	0.1272
DPPV	**0.9534**	-0.0319
KACR	**0.7604**	0.6187
KSS	**0.7591**	0.6165
BGM	**0.7577**	0.6270
DVG	**0.7495**	0.6136
KFPR	**0.6866**	0.4012
DFNR	**-0.6459**	-0.3411
DACR	**-0.9785**	-0.1266
DNPV	0.1114	**0.9864**
KNPV	0.1265	**0.9770**
LGS	0.0667	**0.9371**
DFPR	0.6299	**0.6460**
BRI	-0.5627	**-0.6490**
KGM	0.0632	**-0.9391**
AUC	-0.1737	**-0.9747**
KFNR	-0.1375	**-0.9845**

4.2.1 C4.5N Classifier

The results of the factor analysis of the performance metrics for the C4.5N classifier are shown in Table 4, which shows the rotated factor pattern. 90% of the variance in the performance metrics' space is accounted for by a mineigen value of 1.0, resulting in two factors which group the 22 performance metrics. The first factor groups the KPPV, KFM, DFM, BFM, PRC, DGM, DPPV, KACR, KSS, BGM, DVG, KFPR, DFNR, and DACR performance metrics – shown in bold face for Factor 1 in Table 4. Among these 14 metrics, relatively large factor loadings on the second factor are also observed for KACR, KSS, BGM, and DVG, implying that these four performance metrics are influenced by both factors. The remaining eight metrics (among the 22 total) are associated with the second factor (shown in bold face for Factor 2 in Table 4), as seen in the rotated factor pattern table. Note that, for a performance metric, a negative factor loading implies that an increase in the value of that metric is correlated with a decrease in the values of the performance metrics positively associated with the same factor.

When the mineigen value is changed to 0.5, three factors are obtained, which account for about 95% of the total variance. The consequent rotated factor pattern (not shown) is a bit different than when the mineigen value is 1.0. More specifically, the first factor groups the same performance metrics as previously, except for the DACR metric which is now associated with the second factor. The third factor is associated with only the KFNR metric (i.e., it is a unique factor), which moved from the second factor when mineigen was 1.0.

Table 5. Most Correlated Metrics

C4.5N		C4.5D	
Metrics' Pair	Corr. Coeff.	Metrics' Pair	Corr. Coeff.
BGM/KGM	0.9993	BRI/DACR	-0.9961
KSS/BGM	0.9988	BFM/DFM	0.9918
KSS/KGM	0.9974	DFNR/DFM	-0.9904
BRI/LGS	0.997	KFPR/KACR	-0.9899
BRI/DACR	-0.9958	BFM/PRC	0.9848
DFNR/DFM	-0.9903	BFM/DFNR	-0.9808
BFM/DFM	0.9892	KSS/BGM	0.9802
KPPV/KFM	0.9875	DNPV/DACR	0.9798
LGS/DACR	-0.9871	DFM/DGM	0.9784
BFM/PRC	0.9866	DFM/PRC	0.9768
DFM/DGM	0.9858	BRI/DNPV	-0.9762
KFPR/KACR	-0.9853	BFM/DGM	0.9724
DFM/PRC	0.9843	AUC/KSS	0.9717
DFNR/DGM	-0.9839	BFM/KFM	0.9709
DNPV/DACR	0.9829	BGM/KGM	0.9665
BRI/DNPV	-0.9825	AUC/BGM	0.9654
DPPV/DFM	0.9815	DFNR/PRC	-0.9628
AUC/BGM	0.9804	KPPV/KFM	0.9627
AUC/KGM	0.9797	DFNR/DGM	-0.9616
BFM/DFNR	-0.9789	DFNR/KFM	-0.9594
AUC/KSS	0.9786	KSS/DFM	0.9587
DPPV/PRC	0.9764	DFM/KFM	0.958
BFM/DPPV	0.9736	DPPV/DFM	0.9574

An analysis of the correlation matrix of the performance metrics revealed some notable correlations, which are summarized in Table 5. The results for C4.5N are shown in the first two columns of Table 5, while those for C4.5D are shown in the last two columns of the table. Only the results of C4.5N are discussed in this section, while those of C4.5D are discussed in the next section. The table shows the 23 most correlated pairs (of performance metrics) that fall within the 10^{th} percentile and have a correlation coefficient of between 0.97 and 1.00.

We discuss the top three most correlated pair of performance metrics. The table shows that, for C4.5N, the highest correlation exists between the BGM and KGM metrics. The BGM and KGM metrics respectively represent the geometric mean computed for the decision threshold that maximizes the geometric mean or maximizes the KSS metric. Table 5 also shows that KSS is highly related to both BGM and KGM. Consequently, any metric that is closely related to either BGM or KGM is likely to be also closely related to the other one. A classifier that provides a high value for the KSS metric is able to separate the two classes very well, and both the true positive and true negative rates will be relatively high. These metrics are also the basis for the calculation of the geometric mean. Since the KSS metric is related with BGM and KGM it must also be related with the AUC metric because AUC is closely related to both BGM and KGM – as verified by the table.

4.2.2 C4.5D Classifier

Similar to the results shown for the C4.5N classifier, the corresponding results for the C4.5D classifier are shown in Tables 6 and 5, which respectively show the rotated factor pattern and the most correlated pairs of performance metrics. When the mineigen value is set to 1.0, three factors are obtained as shown in Table 6, which account for about 92% of the variance. When the mineigen value is changed to 0.5 the rotated factor pattern (not shown) shows four factors, accounting for about 95% of the variance.

For the rotated factor pattern shown, the first factor groups the BFM, PRC, DFM, KFM, BGM, AUC, DGM, KSS, DPPV, KGM, KPPV, DVG, KFNR, and DFNR performance metrics, where KFNR and DFNR have negative factor loadings. The second factor groups the DACR, DNPV, KNPV, LGS, DFPR, and BRI performance metrics, while the third factor groups the remaining two metrics, i.e., KACR and KFPR. Among the rotated factor patterns for the C4.5N and C4.5D classifiers, the KPPV, KFM, DFM, BFM, DGM, DPPV, KSS, BGM, DVG, and DFNR performance metrics are associated with the respective first factors.

The most correlated performance metrics according to C4.5D are shown in Table 5 – columns three and four. The most correlated pair involves the BRI and DACR performance metrics. For the instances belonging to class c_0, BRI measures how far the predicted probability of class c_0 membership is from 1, while for an instance belonging to class c_1, BRI measures how far the predicted probability of class c_1 membership is from 1. BRI can also be thought of as the sum of the squared errors obtained by the classifier. Contrast this with DACR, which will be 1 (the perfect model) if $\hat{p}(c = 0 \mid x^i) > 0.5 \; \forall \; x^i$ in class c_0 and $\hat{p}(c = 1 \mid x^i) > 0.5 \; \forall \; x^i$ in class c_1. At the default threshold of 0.5, accuracy and BRI are highly correlated.

5 Conclusion

A study of the relationships between a large number of classifier performance metrics is presented in this work. A collection of 22 performance metrics are collected from C4.5 classifiers (two versions) trained with 35 datasets. The case study datasets suffer from varying degrees of class imbalance. The performance metrics space, for a given classifier, is analyzed using factor analysis. This study is unique in applying factor analysis for evaluating the relationship among commonly used classifier performance metrics.

It was observed that a relatively small number of underlying factors influenced the 22 performance metrics – a result reflecting the fact that most of the 22 metrics are based on the four values obtained from the two-by-two confusion matrix of a classifier. The relationships of the 22 performance metrics were analyzed, and consequently, it is

Table 6. C4.5D: Rotated Factor Pattern

	Factor 1	Factor 2	Factor 3
BFM	**0.9811**	-0.0236	0.1676
PRC	**0.9757**	-0.0002	0.1115
DFM	**0.9751**	0.0355	0.1841
KFM	**0.9582**	-0.1070	0.1552
BGM	**0.9576**	0.1304	0.1263
AUC	**0.9498**	0.1644	0.1357
DGM	**0.9465**	-0.0428	0.2623
KSS	**0.9461**	0.2249	0.1779
DPPV	**0.9258**	0.0172	0.2426
KGM	**0.9181**	0.0631	0.2560
KPPV	**0.8930**	-0.0867	0.1955
DVG	**0.7035**	0.3989	0.0189
KFNR	**-0.8542**	-0.1261	0.4631
DFNR	**-0.9703**	-0.0529	-0.1537
DACR	0.0677	**0.9856**	0.0550
DNPV	0.1401	**0.9551**	0.0761
KNPV	0.1954	**0.8464**	0.2554
LGS	0.0308	**-0.8148**	0.0461
DFPR	0.1889	**-0.9126**	-0.0487
BRI	-0.0966	**-0.9876**	-0.0414
KACR	0.4513	0.2234	**0.8594**
KFPR	-0.3930	-0.1594	**-0.9022**

not recommended to use performance metrics for classifier evaluation that are highly related. Such metrics do not add unique knowledge for evaluating a classifier's performance.

The analyst can use the results presented to determine: (1) which performance metrics should be used to evaluate a classifier, and (2) which performance metrics should not be used together because they are highly related. Only relatively unrelated performance metrics which are associated with different factors and which are not highly related should be used in order to evaluate a given classifier. In the context of our study, the analyst could select one or more performance metrics from each factor (of the rotated factor pattern table) for evaluating a classifier. Referring to Table 6 for example, the analyst could choose one or more performance metrics from the first factor, one or more metrics from the second factor, and one or more metrics from the third factor and come up with metrics AUC (from Factor 1), BRI (from Factor 2), and KACR (from Factor 3) to obtain a performance metrics set that reduces redundancy and covers different aspects of classifier performance.

Future works will include other existing or new performance metrics for classifier evaluation using the explained approach. In addition, other classification algorithms will be considered in the context of this study.

References

[1] R. Alaiz-Rodriguez, N. Japkowicz, and P. Tischer. Visualizing classifier performance on different domains. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*, pages 3–10, Dayton, OH, November 2008. IEEE Computer Society.

[2] A. Asuncion and D. Newman. UCI machine learning repository. *http://archive.ics.uci.edu/ml/*, 2007. Department of Information and Computer Sciences, University of California, Irvine.

[3] R. Caruana and A. Niculescu-Mizil. Data mining in metric space: An empirical analysis of supervised learning performance criteria. In *Proceedings of 10th ACM International Conference on Knowledge Discovery and Data Mining, KDD'04*, Seattle, WA, August 2004.

[4] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 115–123, 1210 West Dayton Street, Madison, WI, 53706 USA, 2006. University of Wisconsin-Madison.

[5] T. Fawcett. An introduction to roc analysis. In *Pattern Recognition Letters*. Elsevier B.V., 2005.

[6] D. J. Hand and W. E. Henley. Statistical classification methods in consumer credit scoring: A review. In *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 521–543, 1997.

[7] B. Hoadley and R. M. Oliver. Business measures of scorecard benefit. *IMA Journal of Mathematics Applied in Business and Industry*, 9:55–64, 1998.

[8] N. Japkowicz. Classifier evaluation: A need for better education and restructuring. In *Proceedings of the 3rd Workshop on Evaluation Methods for Machine Learning, ICML 2008*, Helsinki, Finland, 2008.

[9] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, 5^{th} edition, 2002.

[10] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse. An empirical study of learning from imbalanced data using random forest. In *ICTAI '07: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Vol.2 (ICTAI 2007)*, pages 310–317, Washington, DC, USA, 2007. IEEE Computer Society.

[11] T. M. Khoshgoftaar and N. Seliya. Comparative assessment of software quality classification techniques: An empirical case study. *Empirical Software Engineering Journal*, 9(3):229–257, 2004.

[12] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

[13] J. C. Munson and T. M. Khoshgoftaar. Measuring dynamic program complexity. *IEEE Software*, 9(6):48–55, 1992.

[14] N. J. Pizzi, R. Summers, and W. Pedrycz. Software quality prediction using median-adjusted class labels. In *Proceedings: International Joint Conference on Neural Networks*, volume 3, pages 2405–2409, Honolulu, HI, May 2002. IEEE Computer Society.

[15] SAS Institute. *SAS/STAT User's Guide*. SAS Institute Inc., 2004.

[16] N. Siddiqi. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley, John & Sons, Incorporated, 1st edition, 2005. ISBN 978-0-47-175451-0.

[17] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In *In the Australian Conference on Artificial Intelligence*, pages 1015–1021, 2006.

[18] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine Learning*, pages 935–942, Corvalis, OR, June 2007.

[19] S. Viaene, R. Derring, and G. Dedene. A case study of applying boosting naive bayes to claim fraud diagnosis. In *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society, May 2004.

[20] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Publishers, San Francisco, California, 2nd edition, 2005. ISBN 978-0-12-088407-0.