Cost-sensitive classifier selection when there is additional cost information

Ryan Meekins RMM6EY@VIRGINIA.EDU
Stephen Adams SCA2C@VIRGINIA.EDU
Peter A. Beling PB3A@VIRGINIA.EDU
Department of Systems and Information Engineering
University of Virginia, Charlottesville, VA
Kevin Farinholt FARINHOLTK@LUNAINC.COM
Nathan Hipwell HIPWELLN@LUNAINC.COM
Ali Chaudhry CHAUDHRYA@LUNAINC.COM
Luna Innovations, Charlottesville, VA
Sherwood Polter
Qing Dong
Naval Surface Warfare Center Philadelphia Division, Philadelphia, PA

Editors: Luís Torgo, Stan Matwin, Gary Weiss, Nuno Moniz, Paula Branco

Abstract

Machine learning models are increasing in popularity in many domains as they are shown to be able to solve difficult problems. However, selecting a model to implement when there are various alternatives is a difficult problem. Receiver operating characteristic (ROC) curves are useful for selecting binary classification models for real world problems. However, ROC curves only consider the misclassification cost of the classifier. The total cost of a classification system includes various other types of cost including implementation, computation, and feature costs. To extend the ROC analysis to include this additional cost information, the ROC Convex Hull with Cost (ROCCHC) method is introduced. This method extends the ROC Convex Hull (ROCCH) method, which is used to select potentially optimal classifiers in the ROC space using misclassification cost, by selecting potentially optimal classifiers using this additional cost information. The ROCCHC method is tested using three binary classification data sets, each of which include real feature costs as the additional cost information. Competing classifiers are created with the CART algorithm by using each combination of features or sensors for each data set. The ROCCHC method reduces the classifier decision space to 4%, 9%, and 0.02%. These results are compared to the current ROCCH method, which misses 91%, 58%, and 6% of potentially optimal classifiers because the method does not include the additional cost information.

Keywords: receiver-operating characteristics curves, cost-sensitive learning, cost-sensitive classifier selection, cost-sensitive feature selection

1. Introduction

As cyber-physical systems (CPS) become more prevalent in society, where millions of sensors will be used for managing smart systems such as smart cities, intelligent transportation networks, smart grids, smart homes, etc., ensuring that all of these systems are cost-effective

is paramount (Lee, 2008). CPS rely heavily on machine learning models to make decisions based on their sensor readings. For example, a diagnostic CPS could use sensor information and a classifier to estimate the current health of the system.

Current machine learning methods commonly assume that competing classification models are implemented at an equal cost, however, in real world applications, these systems can have vastly different costs. In the following, "classifier" should be interpreted as a classification system that includes all hardware and software, such as the machine learning algorithm, the feature set, and the feature set's corresponding sensors and tests. Stakeholders selecting a classifier would assess performance as well as operating costs, including the hardware, software, personnel, electricity, etc., of competing classifiers. Current classifier selection methods that include aspects of cost are in the field of cost-sensitive learning.

Cost-sensitive learning is a type of machine learning that takes the costs of misclassifications and other types of cost into account (Ling and Sheng, 2011). The goal of cost-sensitive learning is to minimize the total cost, which consists of the misclassification cost, test or feature cost, computation cost, and all other types of cost (Turney, 2000).

Cost-sensitive classifier selection attempts to select the classifier that solves this optimization problem, the optimal classifier. Cost-sensitive feature selection is a type of cost-sensitive classifier selection that aims to select a classifier that minimizes the feature set cost while still maintaining a high performance. Generally, cost-sensitive classifier selection methods have focused on selecting classifiers that minimize the misclassification cost, while cost-sensitive feature selection methods have focused on minimizing feature set cost and maximizing accuracy. However, a method that incorporates these two ideas into one method that selects classifiers that minimize total cost, including misclassification cost and any additional cost information (i.e. feature set cost), has not been developed.

Our method to accomplish this for binary classification problems utilizes receiver operating characteristics (ROC) curves. ROC analysis is popular because it is robust to imbalanced data sets and unknown costs of misclassification, both of which characterize real world problems (Provost and Fawcett, 1997). The ROC Convex Hull (ROCCH) method has been favored for selecting classifiers in ROC space because it selects a set of potentially optimal classifiers even with the real world costs of misclassification and class distribution unknown. However, this method ignores types of cost other than misclassification cost. Some of these ignored types of cost are often known or can be estimated for real world problems, such as the expected implementation cost of each competing classifier. To extend the ROCCH method to aid in selecting potentially optimal classifiers using this additional cost information, the Receiver Operating Characteristics Convex Hull with Cost (ROCCHC) method is proposed.

2. Background

This section includes current literature in cost-sensitive feature selection along with a review of binary classification, ROC analysis, and the ROCCH method.

2.1. Cost-Sensitive Feature Selection

There are three main ways to perform cost-sensitive feature selection, all of which expand on the traditional feature selection methods of a filter, wrapper, and embedded method.

Filter methods use statistical merit metrics such as p-value to reduce or "filter" the feature set before choosing a classification algorithm. Bolón-Canedo et al. (2014) expanded filter methods to include feature costs using a general cost-sensitive feature selection framework . In this simple framework, feature costs are subtracted from a statistical merit metric to create a new evaluation metric that is cost-sensitive. This subtraction includes a weight parameter placed on the feature cost, requiring stakeholders to determine a tradeoff between the statistical merit metric and the feature cost. Adams et al. (2017a) expanded this framework to include more filtering techniques and machine learning algorithms .

Wrapper methods use an iterative search process of changing a model's feature set and then evaluating the new model's performance. The traditional wrapper algorithms of forward and backward selection sequentially add or remove features from the model by acting greedily with respect to a performance metric, such as accuracy. Wrapper methods have been recently expanded to include feature costs. Kong et al. (2016) added feature costs to a backward selection wrapper by using two evaluation metrics, classification accuracy and feature set cost. The algorithm searches for a minimal feature cost classifier that still achieves a certain accuracy. Min et al. (2014) also used a backward selection wrapper to find a high accuracy classifier that satisfies a maximum feature set cost constraint. Generally, cost-sensitive wrapper methods use a heuristic approach of evaluating accuracy and feature set cost at each iteration in order to decide how to alter the feature set.

Embedded methods select features while building the machine learning model. A common example is a decision tree algorithm, which selects a new splitting feature from the full set using a greedy policy and evaluation metric such as information gain. Embedded methods for cost-sensitive feature selection have been developed recently. Ling et al. (2004) added feature costs and the costs of misclassification to the C4.5 decision tree algorithm by splitting on features that minimize the summation of the feature cost and the misclassification cost. This algorithm is highly efficient, however, the resulting solution may not be globally optimal. Zhou et al. (2016) modified the random forest algorithm by setting the probability that a given feature will be selected as a potential split to the inverse of its cost. This performed cost-sensitive feature selection because high cost features were unlikely to be selected. Adams et al. (2016) developed a cost-sensitive feature selection method for hidden Markov models that simultaneously estimates model parameters and selects features.

The problem with previous work in cost-sensitive feature selection is that a method of selecting potentially optimal classifiers using both misclassification cost and an additional cost, such as the feature set cost, hasn't been developed for real world problems, where the costs of misclassification and class distribution are unknown. The proposed ROCCHC method accomplishes this for binary classification problems.

2.2. Binary Classification

The goal of machine learning classification models is to assign new observations to their actual class. For binary classification models, where new observations can be assigned to either a "Positive" or "Negative" class, Table 1 shows the four possible outcomes. These are a true positive (TP: a "Positive" is correctly classified), a true negative (TN: a "Negative" is correctly classified), a false negative (FN: a "Positive" is incorrectly classified), and a false positive (FP: a "Negative" is incorrectly classified).

Table 1: Binary Classification Outcomes

	Assign Positive	**Assign Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Binary classification models are often evaluated using false positive rate (FPR) and true positive rate (TPR). These are calculated as FPR $= \frac{\#FP}{\#N}$ and TPR $= \frac{\#TP}{\#P}$, where #N is the number of "Negatives" and #P is the number of "Positives" in the test data set. The other terms correspond to the number of a type of outcome. Machine learning classification models assign new observations class membership probabilities based on what is learned from the training observations. For binary classification, an observation's class membership probabilities can be represented in terms of only its "Positive" class membership probability, denoted p_p, (where its "Negative" class membership probability is $1 - p_p$).

A cutoff threshold, $p_{cut} \in [0, 1]$, is used to assign new observations a class membership hypothesis based on the observation's p_p. All observations with $p_p \geq p_{cut}$ are assigned to the "Positive" class and the rest are assigned the "Negative" class. A perfect classifier has a cutoff threshold that corresponds to a FPR $= 0.0$ and TPR $= 1.0$.

You can imagine that the value of p_{cut} greatly influences the FPR and TPR of a given model. Too low of a p_{cut} would result in a high FPR and too high a p_{cut} would result in a low TPR. The special cases of $p_{cut} = 0$ and $p_{cut} = 1$ assign all new observations to either the "Positive" or "Negative" class, respectively. The ROC graph is used to show all FPRs and TPRs for a given classification model as you alter p_{cut} from 0 to 1.

2.3. Receiver Operating Characteristics Curves

The ROC curves is a useful tool for visualizing, organizing, and selecting competing binary classifiers based on their performance. ROC curves have been favored in the machine learning community due to the realization that classification accuracy is an inadequate metric for real world problems, where the costs of misclassification and class distribution are unknown (Provost and Fawcett, 1997).

The ROC curves shows a classifier's operation, in terms of its TPR and FPR, for all p_{cut} from 0 to 1. Figure 1 shows a ROC graph with classifiers, A and B, and a random classifier. The random classifier demonstrates a special case where TPR = FPR for all p_{cut}.

The ideal p_{cut} for a classifier can be determined in ROC space using a line with slope,

$$S = \frac{n \cdot c_{\text{FP}}}{p \cdot c_{\text{FN}}}, \tag{1}$$

where n is the expected number of "Negatives" in the real world, p is the expected number of "Positives" in the real world, c_{FP} is the real world cost of a FP, and c_{FN} is the real world cost of a FN. The ideal p_{cut} for a classifier is found by moving a line with slope S, also shown in Figure 1, from the upper left corner of the ROC graph down and to the right, until it intersects the classifier's ROC curve (Fawcett, 2006).

The point where this line intersects the classifier's ROC curve corresponds to the p_{cut} that would minimize the classifier's total misclassification cost, C_M, which is calculated as

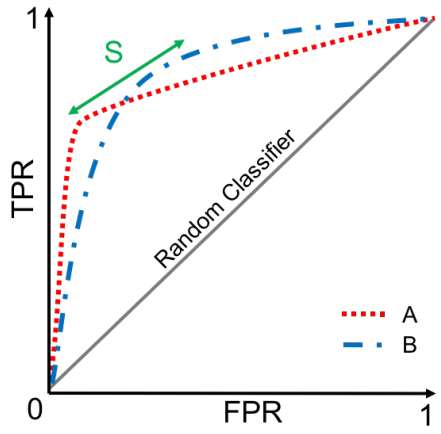

Figure 1: Receiver operating characteristics (ROC) graph

$$C_M = n \cdot \text{FPR} \cdot c_{\text{FP}} + p \cdot (1 - \text{TPR}) \cdot c_{\text{FN}}. \tag{2}$$

This analysis ensures that C_M is minimized, however, the slope, S, can be hard to determine for real world problems because the costs of misclassification (c_{FP} and c_{FN}) and class distribution (p and n) are often unknown (Note that p and n may be different than the collected data's set class distribution, which is usually a real world sample and not the entire population). Provost and Fawcett (1997) have developed the ROC Convex Hull (ROCCH) method to deal with problems where these real world variables are unknown.

The ROC graph has been expanded in other papers, including adding a third dimension of an algorithms ability to detect difficult targets (Alsing et al., 1999) and diagnostic latency for prognostic and health management applications (Simon, 2010). These methods were useful for deciding on alternatives using a 3-D ROC, however, classifiers were not selected using additional cost information.

2.4. ROC Convex Hull Method

The ROCCH method is used to identify a subset of classifiers that are potentially optimal without requiring the real world costs of misclassification or class distribution. This is accomplished using a convex hull in the ROC space, where potentially optimal classifiers have operating points (FPR, TPR) on the convex hull.

The ROCCH analysis uses the concept of an "iso-performance" line, where all operating points in ROC space along a line with slope S will have the same misclassification cost. The "iso-performance" lines closer to the upper-left of the ROC space are more optimal because they correspond to a lower FPR and a higher TPR, resulting in a lower C_M. The ROCCH method selects these classifiers that minimize C_M by using a convex hull in ROC space.

Classifier selection using the ROCCH method is shown in Figure 2. This figure shows a ROC graph consisting of four classifiers, A, B, C, and D. The convex hull of these classifiers is shown using the checkered pattern. Notice that classifiers A and C have operating points

on the convex hull and classifiers B and D do not. This means that only classifiers A and C are potentially optimal. Classifiers B and D would not be chosen for any real world class distribution and costs of misclassification. Further, all operating points of classifiers A and C that are not on the ROC convex hull would never be chosen and could be removed from consideration.

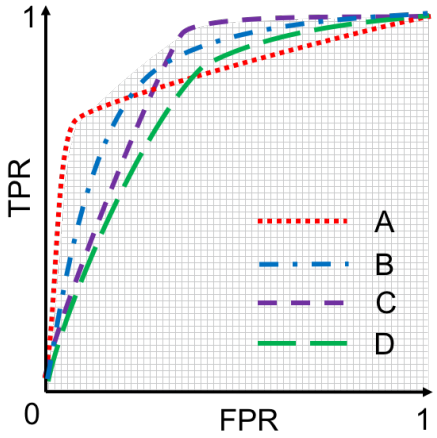

Figure 2: The ROC Convex Hull method shown for four classifiers in an ROC graph. Classifiers A and B are potentially optimal because they have points on the convex hull, which is shown as the checkered pattern.

The ROCCH method finds classifiers that are optimal by minimizing C_M, however, this method ignores other types of cost and, therefore, total cost. For example, classifier B could minimize C if its test cost (or another cost) is lower than that of classifiers A and C. To address this issue, the ROCCHC method is proposed, which adds a new dimension of a known additional cost to the ROC graph.

3. ROC Convex Hull with Cost Method

The ROC Convex Hull with Cost (ROCCHC) method extends the ROC analysis and ROCCH method to aid in real world classifier selection when additional cost information is known. The ROCCHC method provides a solution to the problem that the ROCCH method does not select classifiers that may have a lower total cost when additional cost information is known. The ROCCHC method extends the ROCCH method by adding this additional cost dimension to the ROC graph. This additional cost may be any type of cost other than misclassification cost, such as the expected capital and operating expenses, including the hardware, software, personnel, electricity, etc., for the competing classifiers.

In the following, this additional cost is denoted γ, where γ_i corresponds to this additional cost for the i^{th} classifier. This analysis assumes that γ_i is known for all competing classifiers, where γ could be ordinal (i.e. 1^{st}, 2^{nd}, 3^{rd},...) or numerical. In the proposed method, γ_i is composed of all types of cost other than the misclassification cost and includes but is not limited to the cost of each sensor (which can vary wildly and will influence the cost of each

feature extracted from the sensor), the cost of collecting data, and the cost associated with the power consumed to run the system.

This method builds upon the ROCCH method by using the convex hull in ROC space, however, this method requires computing a convex hull for each unique γ. The ROCCHC method is shown visually in Figure 3. This figure shows classifiers A, B, C, and D again, however, each now has γ shown in parenthesis. There are now two convex hulls, one for each of the unique γ's of \$1 and \$2.

The convex hull calculated at γ of \$1 is shown using the shaded region. This convex hull was computed while including classifiers B and D because their γ is less than or equal to \$1. This convex hull selects classifier B as potentially optimal. Classifier D is not selected as potentially optimal because it's total cost would be greater than classifier B's for any real world costs of misclassification and class distribution.

The convex hull calculated at γ of \$2 is shown using the checkered pattern. This convex hull was computed using all classifiers (A, B, C, and D) because their γ is less than or equal to \$2. This convex hull selects classifiers A and C as potentially optimal.

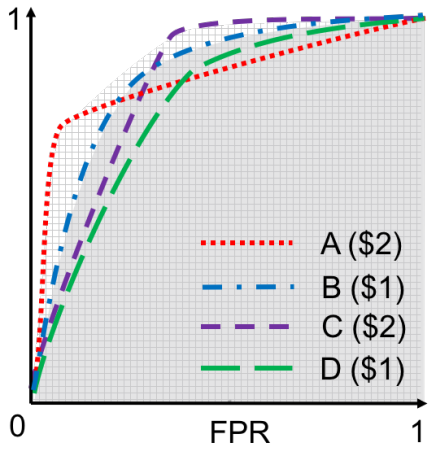

Figure 3: Additional cost information, γ, for each classifier is shown in the legend. The ROCCHC method uses two convex hulls, each at a unique cost, as shown in gray. Classifiers A, B, and C are potentially optimal.

It is important to note that the ROCCHC method will always select at least one classifier at the minimum γ. Also, a classifier i can only be selected as potentially optimal when computing the convex hull for γ_i, which will include all classifiers with a γ equal to or less than γ_i.

The steps to obtain the ROCCHC optimal classifiers from a list of m classifiers is provided in Algorithm 1. The inputs to the algorithm are γ_i and the ROC curve, r_i, of each competing classifier. The number of ROC curve points for each classifier is the number of cutoff thresholds, k. The algorithm outputs the index, i, of each ROCCHC optimal classifier. The algorithm goes to each unique γ and calculates a ROC convex hull of all classifiers with γ_i less than or equal to the current unique γ. A classifier i is ROCCHC optimal if its γ_i equals the current unique γ and its r_i is on the convex hull.

Algorithm 1: ROC Convex Hull with Cost

Input: additional costs: $\gamma_1, \ldots, \gamma_m \in \mathbb{R}$, ROC curves with k cutoff thresholds: $r_1, \ldots, r_m$, where each $r_i \in \mathbb{R}^{2\times k}$ consists of all FPRs $\in \mathbb{R}^{1\times k}$ and TPRs $\in \mathbb{R}^{1\times k}$ of the i^{th} classifier

Output: *Opt*, list of indices of optimal classifiers

```
Opt ← empty list
U ← unique γ sorted in ascending order
for u = 1 to length(U) do
    R ← All r_i corresponding to γ_i ≤ U_u
    H ← Conv(R); where Conv(•) returns i for each classifier r_i in the ROC convex hull
    for i = 1 to m do
        if γ_i = U_u and i ∈ H then
            Append i to Opt
    end
end
```

It is important to mention that the ROCCHC method is not returning all of the classifiers on a 3-D ROC convex hull, where the 3$^{\text{rd}}$ dimension would be γ. The ROCCHC method selects classifiers that outperform all lower cost classifiers (in at least a certain region of ROC space), however, these classifiers are not guaranteed to be in the 3-D ROC convex hull. This was decided so that misclassification cost and γ could have different units or even different number types (cardinal, ordinal, or numeric). For many real world problems, the costs of misclassification (c_{FP} and c_{FP}) may be a relative weighting or are unknown, while γ may be ordinal (i.e. the training computation costs of a competing classification tree and neural network are first and second, respectively). It is our belief that a 3-D convex hull could be used to minimize C only if C_M and γ have the same units (i.e. monetary units).

4. Experiments and Discussion

The ROCCHC method was used to analyze three binary classification data sets, a Pima Native American diabetes data set, a hepatitis data set, and a hydraulic rotary actuator fault detection data set. An ROC analysis is required because the real world costs of misclassification and class distribution are unknown for each data set. These data sets include real test costs for each feature, where the test cost of each feature can be different. The total test cost of a classifier is assumed to be the sum of the classifier's feature set test costs.

The total test cost also includes feature group discounts. Group discounts are applied when groups of features share a common cost. For example, the features from two separate blood tests may have a common cost if the same sample of blood can be used for both tests, resulting in a savings or discount in supplies (i.e. the needle and vial) and personnel (i.e. the nurse). The total test cost of a classifier that includes these two features would be discounted by the common cost.

Table 2 shows the three data sets including the number of features and observations. The test costs for each feature, the feature groups, and feature group common costs are

Table 2: Binary Datasets with Actual Feature Costs

Data set	# of features	# of observations
actuator	56 (8 sensors)	2,340
Pima	8	768
hepatitis	19	155

Algorithm 2: Generation of competing classifiers

for *each combination of features or sensors* **do**
  Get the classifier's test cost based on the combination of features or sensors and feature group discounts
  Build classification tree models using the corresponding feature subset with the CART algorithm and 10-fold cross-validation
  Save the classifier's resulting ROC curve and test cost
end

specified for the Pima Native American diabetes and hepatitis data sets on the UCI Machine Learning Repository (Lichman, 2013). The test costs for the features in the fault detection data set are the actual purchase price of the sensor from which the feature is derived. The feature groups correspond to each feature's required sensor and the common costs are each sensor's actual purchase price. For example, the total test cost of a classifier that includes two features from one sensor will be the purchase price of the sensor and not twice this amount.

For each dataset, competing classifiers were built using each combination of features for the Pima and hepatitis data sets and each combination of sensors for the fault detection data set. The resulting number of classifiers can be determined using $2^b - 1$, where b is the number of features or sensors. Each classifier was created using the CART algorithm (Breiman et al., 1984) in MATLAB with default hyperparameters and validated using 10-fold cross validation. This process is shown in Algorithm 2. The ROCCHC optimal classifiers were then obtained from this list of classifiers using Algorithm 1. These optimal classifiers were compared to ROCCH optimal classifiers for each data set.

4.1. Hydraulic Rotary Actuator Fault Detection Data Set

The proposed framework is demonstrated on a fault detection data set. The objective is to detect faults in a hydraulic rotary actuator given sensor data streams. Each sensor has an associated purchase price as shown in Table 3. More information about this data set can be found in Adams et al. (2017b).

The experimental setup for this data set consists of a hydraulic rotary actuator mounted to another load actuator. The health state of the test actuator can be physically manipulated by artificially inducing several types of faults, such as external leaks, internal leaks, and excessive loadings. There were 2,340 actuation cycles or observations used in this data set, each with a binary fault label.

The 8 sensors include an angular position sensor, 2 flow rate sensors, 2 pressure sensors, and 3 accelerometers. The were 7 features calculated per sensor during each actuation cycle, including mean, variance, standard deviation, skewness, kurtosis, minimum, and maximum.

Table 3: Sensor purchase price for the hydraulic rotary actuator data set

Accelerometer (Accel#)	Angular Position (Angle)	Flow Rate (FlowOut#)	Pressure (PG#)
$35	$40	$3,245	$429

This resulted in 56 (7 feature functions $\times$ 8 sensors) features per observation. For this data set, instead of building classifiers for every combination of features resulting in 7.2×10^{16} $(2^{56} - 1)$ classifiers, classification models were built using every combination of 8 sensors, amounting to 255 $(2^8 - 1)$ classifiers. The test cost of each classifier is the total purchase price of the classifier's sensor set, where sensor purchase prices are shown in Table 3.

The resulting 255 competing classifiers from this data set are shown in Figure 4. This figure shows a 3-D ROC graph, with a third axis of γ as the sensor set cost of each classifier, along with a corresponding ROC graph. The ROCCHC optimal classifiers have solid ROC curves and the sub-optimal classifiers have dotted line ROC curves. This figure shows that increasing γ doesn't necessarily lead to classifiers with better ROC performance. The 3-D ROC graph shows three sensor cost groups, where all 11 ROCCHC optimal classifiers are in the cheapest group.

These 11 ROCCHC optimal classifiers are shown alone in Figure 5, including a legend with each classifier's sensor set cost and sensor set. As expected, the ROCCHC optimal classifiers with a higher γ outperform the others. These 11 classifiers are potential optimal and could be presented to stakeholders, resulting in a 96% $((1 - 11/255)$ reduction in classifiers.

Comparing to the ROCCH method, the ROCCH optimal classifiers were also computed. The only ROCCH optimal classifier is the "$868: PG1, PG2" classifier, which is the ROCCHC optimal classifier with the highest γ. Therefore, ignoring other types of cost (sensor set cost in this case) by using the ROCCH method would miss 10 potentially optimal classifiers, or 91% (10/11).

4.2. Diabetes of Pima Native Americans Data Set

The Pima Native Americans data set consists of 8 features so there are 255 (2^8-1) competing classifiers. The ROCCHC method results in 24 potentially optimal classifiers, which are shown in Figure 6. This figure shows a 3-D ROC graph, with a third axis of γ as the feature set cost of each classifier, along with a corresponding ROC graph. These 24 potentially optimal classifiers could be presented to stakeholders, resulting in a 91% $(1 - 24/255)$ reduction in classifiers.

The ROCCH method results in 10 potentially optimal classifiers. Therefore, ignoring test cost by using the ROCCH method would miss 14 potentially optimal classifiers, which is 58% (14/24) of the potentially optimal classifiers.

4.3. Hepatitis Dataset

The hepatitis data set consists of 19 features so there are 524,287 $(2^{19} - 1)$ competing classifiers. The ROCCHC method results in 84 potentially optimal classifiers, which are shown in Figure 7. This figure shows a 3-D ROC graph, with a third axis of γ as the

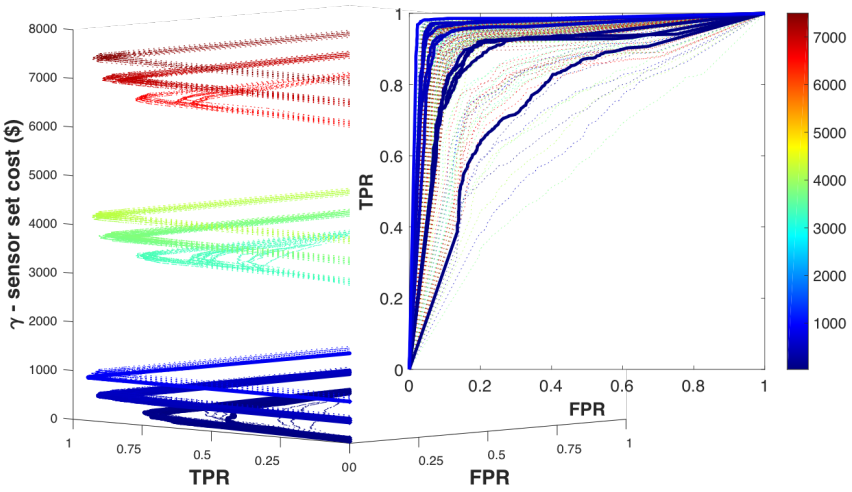

Figure 4: All competing classifiers for the actuator data set (left: 3-D ROC graph with γ as sensor set cost, right: corresponding ROC graph). ROCCHC potentially optimal classifiers have solid line ROC curves, sub-optimal classifiers have dotted line ROC curves.

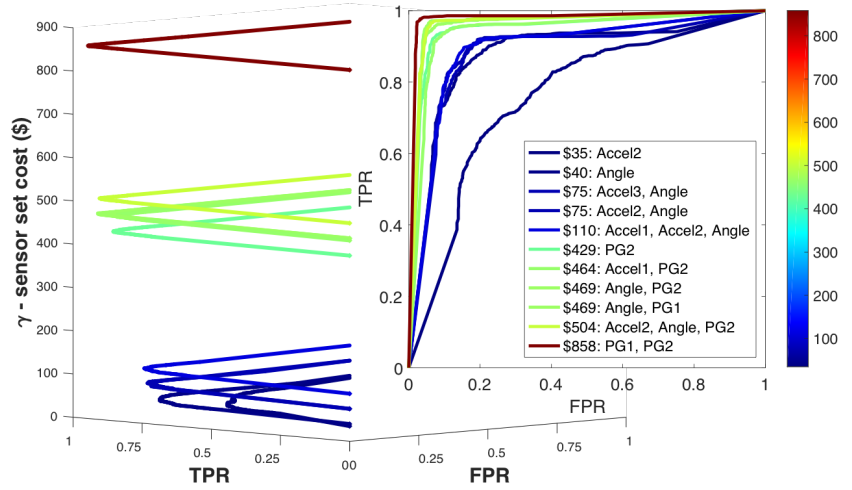

Figure 5: ROCCHC potentially optimal classifiers for the actuator data set (left: 3-D ROC graph with γ as sensor set cost, right: corresponding ROC graph).

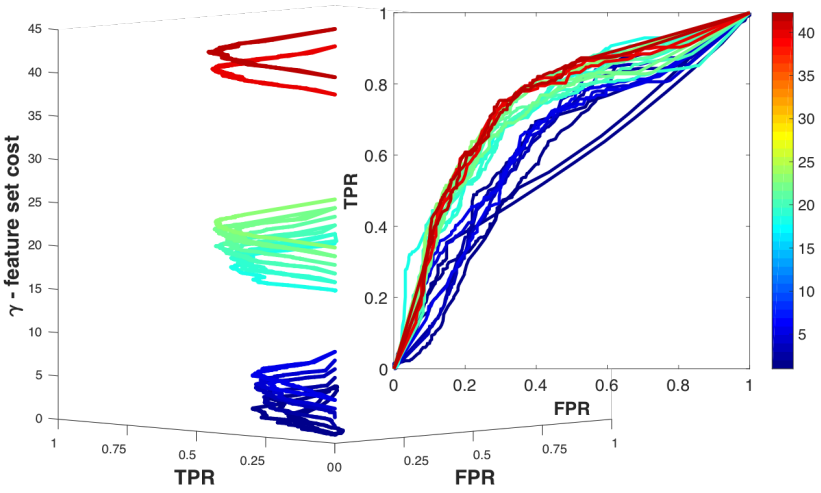

Figure 6: ROCCHC potentially optimal classifiers for the Pima data set (left: 3-D ROC graph with γ as feature set cost, right: corresponding ROC graph).

feature set cost of each classifier, along with a corresponding ROC graph. The 84 ROCHC potentially optimal classifiers results in a 99.98% $(1 - 84/524,287)$ reduction in classifiers.

The ROCCH method results in 79 potentially optimal classifiers, therefore, five potentially optimal classifiers are missed, amounting to 6% (5/84) of the potentially optimal classifiers.

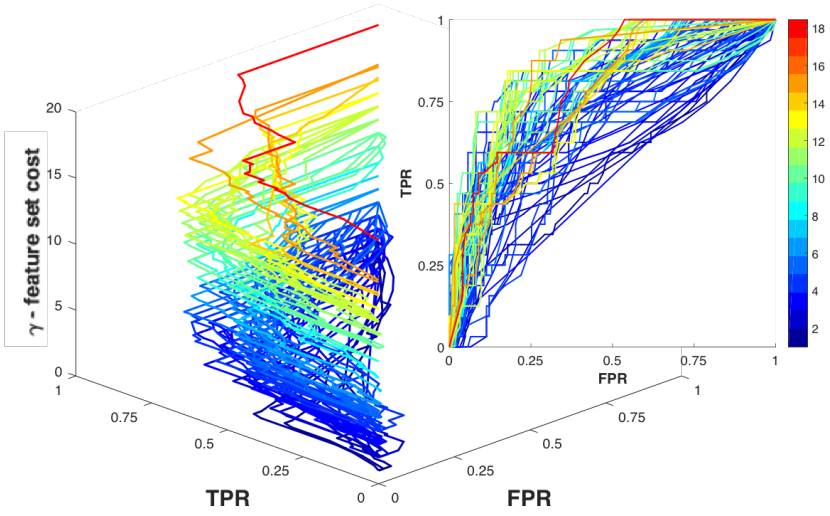

Figure 7: ROCCHC potentially optimal classifiers for the hepatitis data set (left: 3-D ROC graph with γ as feature set cost, right: corresponding ROC graph).

5. Conclusions and Future Work

In conclusion, this study demonstrates the effectiveness of the ROCCHC method at incorporating an additional cost into binary classifier evaluation. This method adds a third dimension to the ROC graph that represents the additional cost. An algorithm for determining ROCCHC potentially optimal classifiers is presented. The numerical experiments demonstrate that the ROCCHC method successfully reduces the number of competing classifiers and allows a stakeholder to evaluate classifiers in terms of the additional cost and ROC performance in a compact visualization. The experiments also compare the presented ROCCHC method to the ROCCH method, which misses 91%, 58%, and 6% of the potentially optimal classifiers. This is because the ROCCH method doesn't incorporate the additional cost information.

There are numerous avenues for possible future work and extensions of the ROCCHC framework. Fawcett (2006) mentions that ROC curves can only be compared if there is a measure of variance, therefore, future methods could use repeated trials of building classifiers, which could be used to determine a confidence interval representing variance. The ROCCHC algorithm would need be updated to account for this ROC confidence interval.

The ROCCHC method could also be used in combination with a search algorithm, such as a forward or backward sequential search, particle swarm optimization algorithm, or genetic algorithm, to find potentially optimal classifiers. This would eliminate the need for the exhaustive feature set search in Algorithm 2. The next steps for this method could also include multi-class problems, where the convex hull would need to include multiple ROC graphs.

Acknowledgments

This material is based upon work supported by the Naval Sea Systems Command under Contract No N00024-17-C-4008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Naval Sea Systems Command.

References

Stephen Adams, Peter A. Beling, and Randy Cogill. Feature Selection for Hidden Markov Models and Hidden Semi-Markov Models. *IEEE Access*, 4:1642–1657, 2016.

Stephen Adams, Ryan Meekins, and Peter A Beling. An Empirical Evaluation of Techniques for Feature Selection with Cost. In *International Conference on Data Mining (ICDM)*, New Orleans, LA, 2017a.

Stephen Adams, Ryan Meekins, Peter A Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. A Comparison of Feature Selection and Feature Extraction Techniques for Condition Monitoring of a Hydraulic Actuator. In *Annual Conference of the Prognostics and Health Management Society*, St. Petersburg, FL., 2017b.

Stephen G. Alsing, Erik P. Blasch, and Kenneth W. Bauer, Jr. Three-dimensional receiver operating characteristic (ROC) trajectory concepts for the evaluation of target recognition

algorithms faced with the unknown target detection problem. volume 3718, pages 449–458. International Society for Optics and Photonics, 8 1999.

V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos. A framework for cost-based feature selection. *Pattern Recognition*, 47(7):2481–2489, 2014.

Leo. Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and regression trees*. Wadsworth, Belmont, CA, 1984. ISBN 0412048418.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, (27):861–874, 2006.

R Ganggang Kong, Liangxiao Jiang, and Chaoqun Li. Beyond accuracy: Learning selective Bayesian classifiers with minimal test cost. *Pattern Recognition Letters*, 80:165–171, 2016.

Edward A. Lee. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 5 2008.

M. Lichman. UCI Machine Learning Repository, 2013. URL http://archive.ics.uci.edu/ml.

Charles X. Ling and Victor S. Sheng. Cost-Sensitive Learning. In *Encyclopedia of Machine Learning*, pages 231–235. Springer US, Boston, MA, 2011. doi: 10.1007/978-0-387-30164-8{_}181. URL http://www.springerlink.com/index/10.1007/978-0-387-30164-8_181.

Charles X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *Twenty-first international conference on Machine learning - ICML '04*, page 69, New York, New York, USA, 2004. ACM Press.

Fan Min, Qinghua Hu, and William Zhu. Feature selection with test cost constraint. *International Journal of Approximate Reasoning*, 55(1):167–179, 1 2014.

Foster Provost and Tom Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 43–48. AAAI, 1997. URL https://www.nssl.noaa.gov/users/brooks/public_html/feda/papers/ProvostandFawcettKDD-97.pdf.

Donald L Simon. A Three-Dimensional Receiver Operator Characteristic Surface Diagnostic Metric. In *Annual Conference of the Prognostics and Health Management Society*, 2010.

Peter D. Turney. Types of Cost in Inductive Concept Learning. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, 2000. URL http://arxiv.org/abs/cs/0212034.

Qifeng Zhou, Hao Zhou, and Tao Li. Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features. *Knowledge-Based Systems*, 95(C):1–11, 3 2016.